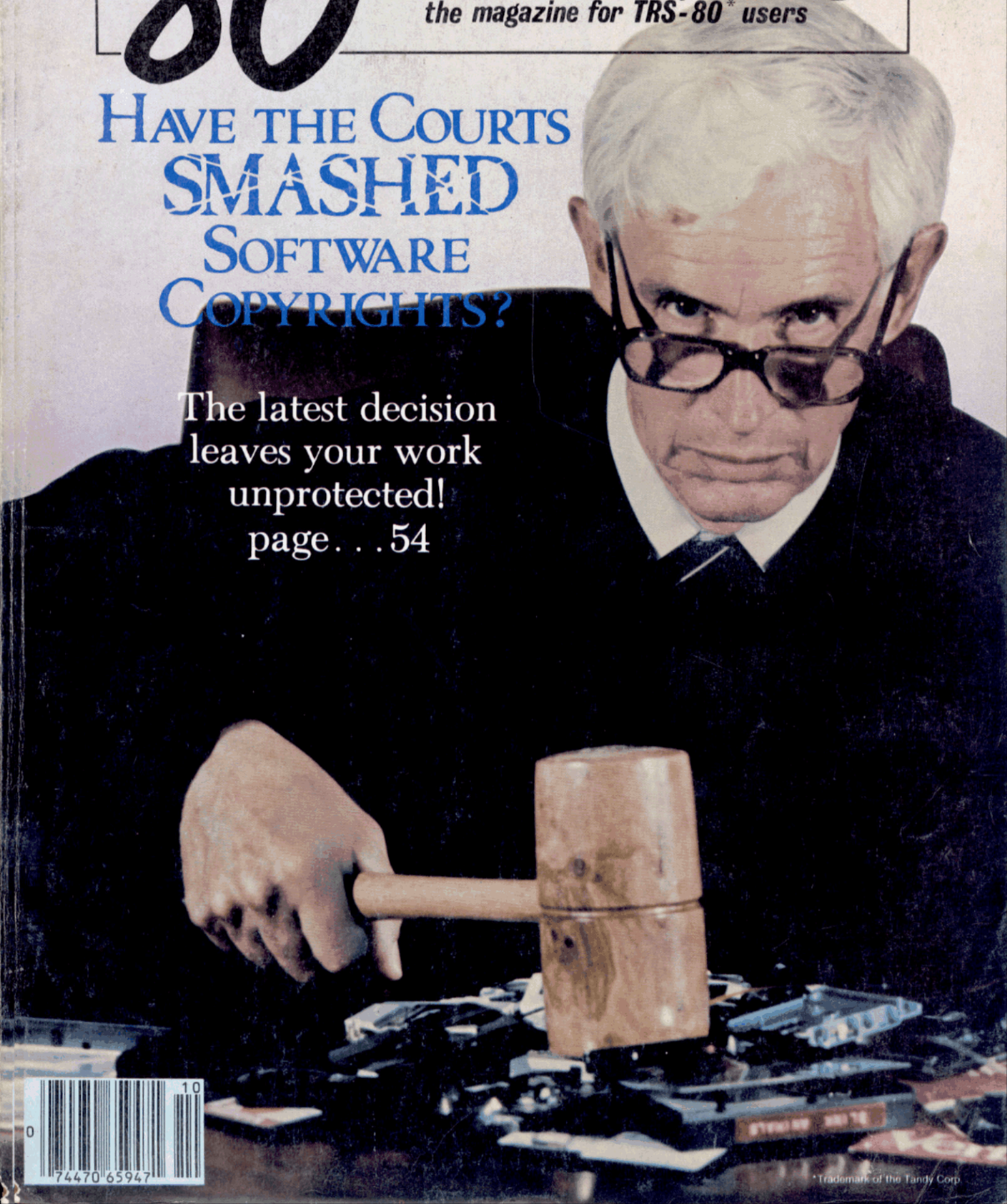
October 1980 $2.50 U.S.

80 microcomputing™

the magazine for TRS-80 users*

HAVE THE COURTS SMASHED SOFTWARE COPYRIGHTS?

The latest decision leaves your work unprotected!
page . . . 54

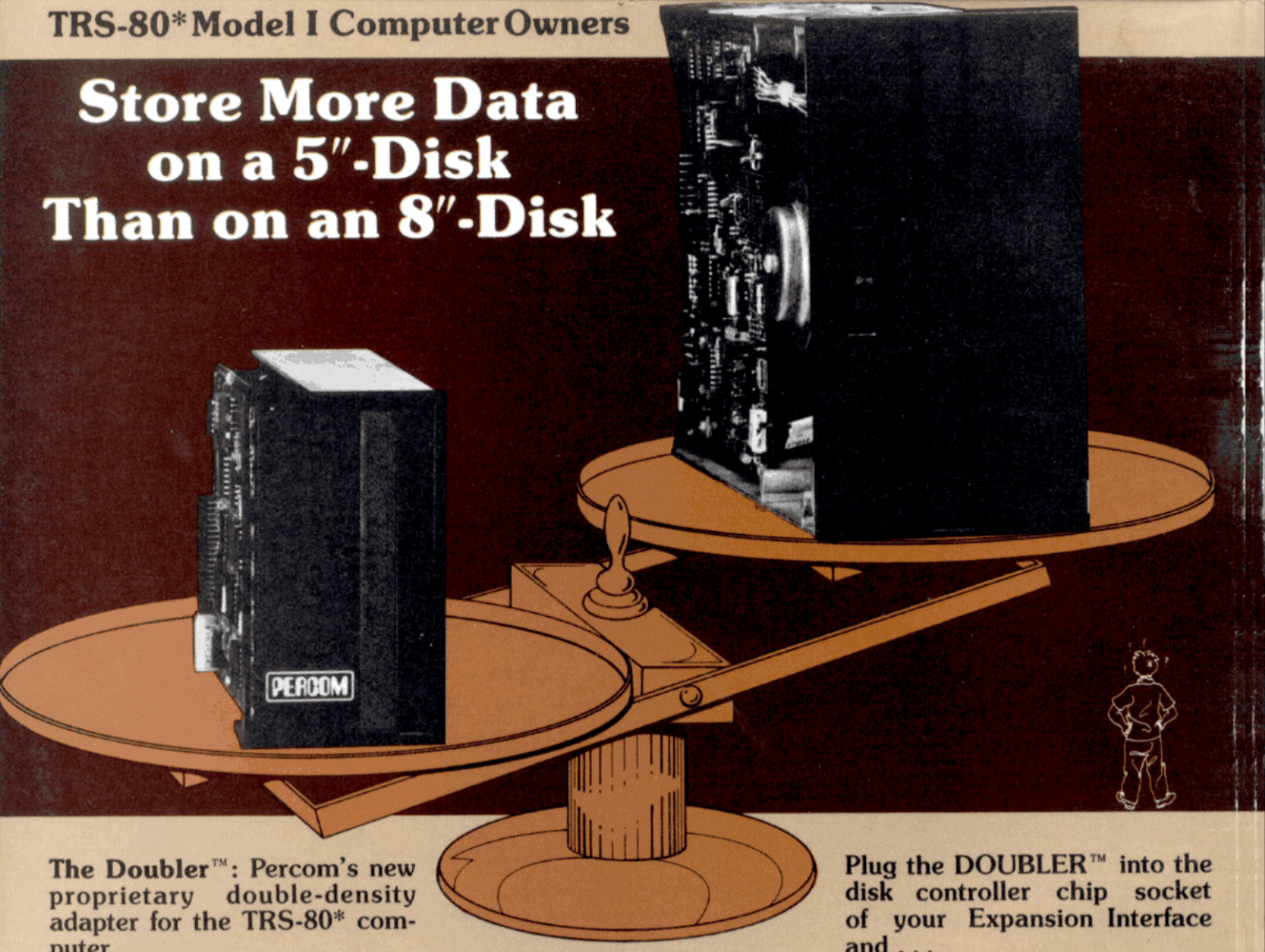


0 74470 65947 10

*Trademark of the Tandy Corp

TRS-80* Model I Computer Owners

Store More Data on a 5″-Disk Than on an 8″-Disk



The Doubler™: Percom's new proprietary double-density adapter for the TRS-80* computer.

Plug the DOUBLER™ into the disk controller chip socket of your Expansion Interface and . . .

Store up to 354 Kbytes of formatted data on five-inch disks.†

- Increase **formatted** storage capacity of your minidiskettes from 1½ to almost 4 times.
- Use with standard 5-inch drives rated for double-density operation.
- The DOUBLER™ reads, writes and formats **either** single- or double-density disks.
- Proprietary design allows you to continue to run TRSDOS*, NEWDOS‡, Percom OS-80™ or other single-density software **without making any changes** to software or hardware.

Mini-Disk Systems
More storage capacity, higher reliability — from Percom, the industry leader. One-, two- and three-drive configurations in either 40- or 77-track format, starting at only $399.

- Includes DBLDOS,™ a **TRSDOS* compatible** double-density disk operating system.
- CONVERT utility, on DBLDOS™ minidiskette, converts files and programs from single- to double-density or double- to single-density.
- **Plug-in installation:** No strapping. No trace cutting. Restore your Expansion Interface disk controller to original configuration by simply removing the DOUBLER™ and reinstalling the original disk controller chip.

PERCOM DISCOUNT COUPON
worth $20
toward
The Purchase of a
DOUBLER™
Coupon No. 80M103
Expires December 30, 1980
Void where prohibited by law.

- The DOUBLER™ circuit card includes high-performance data separator, write precompensation circuits for reliable disk read operations — even on 77-track drives.

Introductory price, including DBLDOS™ and format conversion utility on minidiskette, **only** $219.95. Use the coupon for even greater savings.

Call toll-free, 1-800-527-1592, for the address of your nearest dealer, or to order direct from Percom.

†Percom TFD-200™ drive, OS-80D™ operating system

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

v-1

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

The easiest, least expensive way to generate spectacular multi-color graphics, sharp two-color alphanumerics: Your computer, a color tv set and the Percom Electric Crayon™.

Add the Electric Crayon™ to your system and your keyboard becomes a palette, the tv screen your medium.

You dab and stroke using one-key commands to create dazzling full-color drawings, eye-catching charts and diagrams.

Or you run any of innumerable programs. **Your own BASIC language programs** that generate dynamic pyrotechnic images, laugh-provoking animations.

From a combined alphanumerics-semigraphics mode to a high resolution 256- by 192-element full graphics mode, the microprocessor-controlled Electric Crayon™ is capable of generating **10 distinctly different display modes.**

Colors are brilliant and true, and up to eight are available depending on the mode.

As shipped, the Electric Crayon™ interfaces a TRS-80* computer via your Expansion Interface or Printer Adapter. It may be easily adapted for interfacing to any computer or to an ordinary parallel ASCII keyboard.

But that's not all

The Electric Crayon is not just a color graphics generator/controller.

It is also **a complete self-contained control computer.** With built-in provision for 1K-byte of on-board program RAM, an EPROM chip for extending EGOS™, its on-board ROM graphics OS, and a dual bidirectional eight-bit port — over and above the computer/keyboard port — for peripherals. The applications are endless.

Shipped with EGOS™, 1K-byte of display memory and a comprehensive user's manual that includes an assembly language listing of EGOS™ and listings of BASIC demo programs, the Electric Crayon™ **costs only $249.95.**

Options include:

- LEVEL II BASIC color graphics programs on minidiskette: $17.95.
- A 34-conductor ribbon cable to interconnect the Electric Crayon™ to a TRS-80*: $24.95.
- RAM chips for adding refresh memory for higher density graphics modes: $29.95 per K-byte.
- Electric Crayon™ Sketchpad, a sketching grid of proportioned picture elements (pixels) in a tv aspect ratio. For 128 x 192 or 256 x 192 graphics modes. 11-inch by 17-inch, 25-sheet pads: $3.95 per pad.

SYSTEM REQUIREMENTS: the video circuitry of the Electric Crayon™ provides direct drive input to a video monitor or modified tv set. An internal up-modulator for rf antenna input may be constructed by adding inexpensive components to the existing video circuitry.

Prices and specifications subject to change without notice.

™ = trademark of Percom Data Company, Inc.

* = trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Get into computer color graphics the easy, low-cost way with a Percom Electric Crayon™. Available at Percom dealers nationwide. Call toll-free, 1-800-527-1592, for the address of your nearest dealer, or to order direct if there is no Percom dealer in your area.

META TECHNOLOGIES

FOR YOUR TRS-80® DISK SYSTEM

PROGRAMMING TOOLS

SAVE Any 3, $49.95

For Model II . . . . . $ 74.95

TDAM $19.95
For Model II $29.95
Includes MTC QUE Card!

Having trouble with RANDOM FILES? With MTC's Table-Driven Access Method (TDAM) you'll never fret over FIELDing again. No knowledge of random access files is required. Insert the TDAM "interpreter" into any BASIC program and type in a few DATA statements describing the information in your files. TDAM does the rest! Reads and writes fields and records of any type (even compresses a DATE field into 3 bytes!). Features automatic file buffer allocation/deallocation, memory buffering, sub-record blocking/deblocking, and handles up to 255 fields per record. Super fast and super simple! Complete with TDAM interpreter, instructions and demo program. Requires programming experience.

SIFTER $19.95
For Model II $29.95

Twelve in-memory high-speed sorts for use in any BASIC program: stable, non-stable, with/without tags, for numeric or string data. Random File Sort included. Some sorts written in machine code. Includes sort subroutines, demo programs and instructions. Relocate as needed with REBUILD. Requires programming experience.

SHRINK $19.95
For Model II $29.95

Makes Every Byte Count! Make programs smaller and faster! Combines lines & removes unnecessary code including remarks, without altering program operation. Typically reduces program size 25% to 40%.

DIVERGE $19.95
For Model II $29.95

Compares two BASIC program files, showing the differences between them. Identifies & lists lines which have been inserted, deleted, & replaced. Use for version control.

SUPERSEDE $19.95
For Model II $29.95

A "must have" for the professional programmer or the serious amateur. Probably one of the greatest time-savers available. Write programs in shorthand - change variable names - generate program documentation - use with REBUILD and MINGLE to build new programs from old ones.

REBUILD $19.95
For Model II $29.95

Reorganize programs for adding program code, faster execution, readability. Much more than simple renumbering. Rearrange groups of statements within a program - automatically updates references to line numbers. Use with SUPERSEDE and MINGLE for maximum effect.

MINGLE-II $19.95
For Model II $29.95

Merge up to 14 files (Program or Data) into a single file. Data files may be merged in ascending or descending sequence with the ordering based on a user-specified comparison field. A very handy utility for consolidating data files.

Single sided, Single density, Soft-sectored

DISKETTES

Verbatim 5¼-inch

$23.95 Box of 10

10 Boxes of 10 . . . . . (each box) $22.95
Hard-sectored (10-hole), Box of 10 . . . . $26.95

8-inch FLOPPIES
Single-density, Box of 10 . . . . . $29.95
Double-density, Box of 10 . . . . . $39.95

PLASTIC LIBRARY CASES
5¼-inch or 8-inch diskette case $3.00
50 (5¼-inch) diskette file box $29.95

FACTORY FRESH, ABSOLUTELY FIRST QUALITY. Minimum order 1 box, NO order limit!

NEWDOS/80

by Apparat

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!). Includes 180-page instruction manual and MTC QUE card.

NEWDOS/80 . . . . . $ 149.95
MTC QUE Card only $7.50

CALL REGARDING OUR NEWDOS+ UPGRADE PRICING.

Complete for Model I with all utilities
Plus exclusive MTC QUE card!

NEWDOS + $69.95 by Apparat

40 TRACK VERSION . . . . . $ 79.95

includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

MTC QUE Card only $1.50

The perfect supplement for your NEWDOS, from IJG, Inc.

"TRS-80 DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

132 pages written in PLAIN ENGLISH packed with HOW TO information with details, examples and in-depth explanations. Recover lost files and directories, remove file protection, make BASIC programs unlistable. How to use SUPERZAP, recover from DOS errors and MORE!

TRS-80 DISK $19.95

MORE ➡
PRODUCTS ON PAGES 6 & 7

All products guaranteed for replacement only. Prices, Specifications & Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

QUANTITY DISCOUNT INQUIRIES INVITED

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for standard UPS shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

TO ORDER CALL TOLL FREE
1-800-321-3552
IN OHIO call (216)289-7500 (COLLECT)

META TECHNOLOGIES CORPORATION ✓405

26111 Brush Avenue, Euclid, Ohio 44132

master charge
VISA

800827

October 1980 Issue #10

80 microcomputing T.M. DATA

80 Microcomputing (ISSN #0199-6789) is published monthly by 1001001 Inc., 80 Pine Street, Peterborough, NH 03458. Application to mail second class postage rate is pending at Peterborough, NH 03458 and at additional mailing offices. Phone: 603-924-3873. Subscription rates in the U.S. are $18 for one year and $45 for three years. In Canada, $20—one year only, U.S. funds. Foreign subscriptions (surface mail), $28—one year only, U.S. funds. Foreign subscriptions (air mail), $60—one year only, U.S. funds. In Europe please contact Monika Nedela, Markstr. 3, D-7778, Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, S. Africa 2146. Australian Distributor: Electronic Concepts, Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form #3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.

META TECHNOLOGIES

MTC AIDS-III*

MODEL I . . .$69.95 MODEL II . . .$99.95

Introducing the latest addition to MTC's family of data management systems, AIDS-III. NO PROGRAMMING, easy to use. COMPLETE PACKAGE including demonstration application, documentation and MAPS-III (see below).

- Up to 20 USER-DEFINED FIELDS of either numeric- or character-type.
- CHARACTER-type fields may be any length (total: up to 254 characters).
- NUMERIC-type fields feature automatic formatting, rounding, decimal alignment and validation.
- Full feature EDITING when adding or changing records:
  - ENTER FIELD (can't type-in more characters than specified).
  - BACKSPACE (delete last character typed).
  - DELETE FIELD contents.
  - RESTORE FIELD contents.
  - RIGHT-JUSTIFY FIELD contents.
  - SKIP FIELD (to next or previous field).
  - SKIP RECORD (to next or previous record).
- SORTING of records is MACHINE CODE assisted.
  - 200 RECORDS (40 characters) in about 5 SECONDS.
  - ANY COMBINATION of fields (including numerics) with each field in ascending or descending order.
- SELECTION of records for Loading, Updating, Deleting, Printing and Saving is MACHINE CODE assisted.
  - Specify up to 4 CRITERIA, each using one of 6 RELATIONAL COMPARISONS.
  - LOAD or SAVE selected records using MULTIPLE FILES.
  - Example: Select records representing those people who live in the state of Colorado, but not in the city of Denver, whose last names begin with "F" and whose incomes exceed $9000.00.
  - Example: Select records representing those sales made to XYZ COMPANY that exceed $25.00, between the dates 03/15 and 04/10.

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge, has the following features:

- Full AIDS-III SELECTION capabilities.
- Prints user-specified fields DOWN THE PAGE.
- Prints user-specified fields in titled, columnar REPORT FORMAT, automatically generating column headings, paging and (optionally) indentation.
- Can create a single report from MULTIPLE FILES.
- Prints user-defined formats for CUSTOM LABELS, custom forms, etc.

BELOW ARE TESTIMONIALS from owners of AIDS systems. These are absolutely authentic statements and are typical of the comments we receive.

"This program will do more for my business than all the other programs I have, combined."
David Wareham, Vice President (EDP), National Hospital and Health Care Services Inc.

"We have 32 different Data Base Management packages for the TRS-80. AIDS-III is easily the best. It also makes it easier for us to step up to our Model II since the package is available for both computers."
Jack Bilinski, President, 80 Microcomputer Services

"Your AIDS program is far and away the finest information management system that I've ever seen. I am currently using it to maintain a clear picture of the demographic data on all the kids in our residential treatment program and it is working for me superbly."
Frank Boehm, Director, Front Door Residential Treatment Program

- COMPATIBLE with AIDS-II data files and AIDS subsystems.
- Move up from AIDS-II and EXPAND to 20 field capability WITHOUT REENTERING DATA.
- AIDS-II (Model I or II) owners may UPGRADE FOR ONLY $25.00.

*WARNING! This program is written in BASIC and can be listed in the normal manner. Modification of program code is NOT RECOMMENDED due to its extreme complexity.

MTC AIDS - II

Ailing information? Doctor it up with AIDS-II. This Automated Information Directory System offers twelve user-defined fields with full feature editing when adding or changing records. Selective Loading, Updating, Deleting, Printing and Saving records may be accomplished using any of six relational comparisons. Also features machine code assisted sorting (200 records in about 5 seconds) by any combination of fields, and much more! Unique "windowing" capability allows directories of unlimited size. Window size is typically 200 or more records in 32K. Can be used for mailing lists, client reference reporting, appointment "calendars", inventory records and other information systems. Easy to use. Defining a system takes about a minute. MAPS-I (MTC AIDS PRINT SUBSYSTEM) is included at no charge. MAPS features full AIDS-II selection capabilities, prints user-specified fields down the page, produces user-specified columnar report formats with automatically generated column headings and paging, and allows user-defined print formats for custom forms, labels, etc. Add subsystems for additional capabilities. May be upgraded to AIDS-III when required.

MTC AIDS-II . . . . . $ 49.95
For Model II . . . . . $ 79.95

AND OTHER MYSTERIES

Volume II
Forward by H. C. Pennington

Call now and place your order for his new book, "MICROSOFT™ BASIC AND OTHER MYSTERIES." A primer for cassette and disk BASIC on the TRS-80, the information provided applies to similar MICROSOFT™ BASIC interpreters. Features include definition of terms, an overview of BASIC and DOS, explanation of exits, error codes, verb actions, "cold" and "warm" restart procedures, and examination of system utilities, arithmetic support and I/O driver routines, and the communications region in RAM. Individual routines are explained in detail, with an index provided for easy access. Appendixes include tables for BASIC and DOS vectors, stacks and interrupt locations, PLUS thousands of comment lines for the complete MICROSOFT™ BASIC. Available from the publisher in just a few short weeks, the price is less than $30.

← MORE
PRODUCTS ON PAGE 4

Let your TRS-80® Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80® Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 . . . . . $69.95

Let Your TRS-80® Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 . . . . . $29.95

Let Your TRS-80® Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

A complete checkup for your Model I. THE FLOPPY DOCTOR completely checks every sector of 35- or 40-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

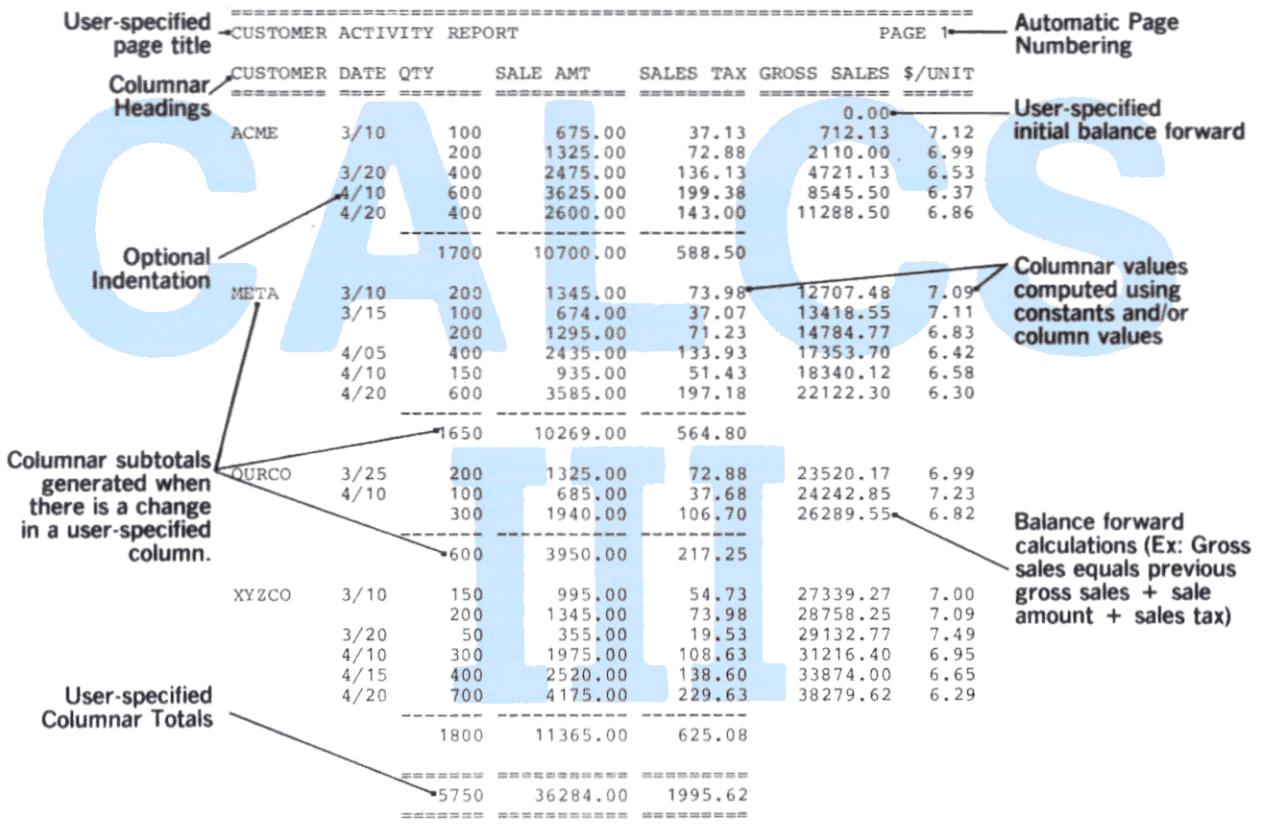
SYSTEM DIAGNOSTICS $19.95

MAKES EVERY BYTE COUNT

IN YOUR TRS-80® MODEL I OR MODEL II DISK SYSTEM

MTC AIDS CALCULATION SUBSYSTEM-III

MODEL I . . .$24.95 MODEL II . . .$39.95




```
CUSTOMER ACTIVITY REPORT                                        PAGE 1

CUSTOMER  DATE  QTY      SALE AMT     SALES TAX  GROSS SALES  $/UNIT
========  ====  =======  ==========   =========  ===========  ======
                                                        0.00
ACME      3/10      100       675.00      37.13       712.13    7.12
                    200      1325.00      72.88      2110.00    6.99
          3/20      400      2475.00     136.13      4721.13    6.53
          4/10      600      3625.00     199.38      8545.50    6.37
          4/20      400      2600.00     143.00     11288.50    6.86
                  -------  ----------   ---------
                   1700     10700.00     588.50

META      3/10      200      1345.00      73.98     12707.48    7.09
          3/15      100       674.00      37.07     13418.55    7.11
                    200      1295.00      71.23     14784.77    6.83
          4/05      400      2435.00     133.93     17353.70    6.42
          4/10      150       935.00      51.43     18340.12    6.58
          4/20      600      3585.00     197.18     22122.30    6.30
                  -------  ----------   ---------
                   1650     10269.00     564.80

OURCO     3/25      200      1325.00      72.88     23520.17    6.99
          4/10      100       685.00      37.68     24242.85    7.23
                    300      1940.00     106.70     26289.55    6.82
                  -------  ----------   ---------
                    600      3950.00     217.25

XYZCO     3/10      150       995.00      54.73     27339.27    7.00
                    200      1345.00      73.98     28758.25    7.09
          3/20       50       355.00      19.53     29132.77    7.49
          4/10      300      1975.00     108.63     31216.40    6.95
          4/15      400      2520.00     138.60     33874.00    6.65
          4/20      700      4175.00     229.63     38279.62    6.29
                  -------  ----------   ---------
                   1800     11365.00     625.08

                  =======  ==========   =========
                   5750     36284.00    1995.62
                  =======  ==========   =========
```

Compare AIDS-III/CALCS with any other data management package under $100!

Others make claims, CALCS-III delivers with user-specified:

- Fields in any order, with optional indentation
- Computations using field values and constants
- Columnar subtotals and totals
- Full AIDS-III selection of records to be printed

Use for accounting, inventory, financial and other numeric-based information packages.

CALCS-III REQUIRES THE PURCHASE OF AIDS-III OR AIDS-II

All products guaranteed for replacement only. Prices, Specifications & Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

QUANTITY DISCOUNT INQUIRIES INVITED

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for standard UPS shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

TO ORDER CALL TOLL FREE

1-800-321-3552

IN OHIO call (216)289-7500 (COLLECT)

20

META TECHNOLOGIES CORPORATION

26111 Brush Avenue, Euclid, Ohio 44132

800827

TRS-80 and Radio Shack are registered trademarks of Tandy Corp.

80 REMARKS
by Wayne Green

"I can't really be critical of Radio Shack for being slow in keeping up with the pioneers of new hardware and improvements on their software."

What About Tandy?

You'll find me being critical of Radio Shack where I feel that they can improve their act. I think you'll find me properly appreciative when they do come up with improvements and, hopefully, tolerant when I can see that things are beyond their control.

For instance, I can't really be critical of Radio Shack for being slow in keeping up with the pioneers of new hardware and improvements on their software. Since I am fighting a similar battle against time with Instant Software, I realize how long it takes to implement something new, particularly when your reputation is at stake. It's easy to rush a new program out to the market as long as you don't care whether it has bugs or not, or whether it is the best one out of its type. This is why so many of the smaller program houses have such a high percentage of crap. Tandy can't afford that any more than Instant Software. Innovation takes an exasperatingly long time, and we live with it, though not graciously.

Recognizing that I tend to hear the horror stories, I'd like to hear from any programmers who have had a happy relationship with Radio Shack. Before I go warning programmers to be extra careful, I'd like to make sure that I have the facts. Billion dollar firms are difficult to deal with and can inadvertently squash individuals without being aware of it. The higher ups are protected from the flak by armies of lower echelon people, who do not want to "bother" the bosses.

Change in Strategy

When *80* was started, I planned to keep the higher level TRS-80 articles in *Kilobaud Microcomputing* as a sort of "next step upwards" for computerists. Since that time, the market has changed and the magazines must change with it. *KM* has gained a wider business and education-oriented readership.

There are still hobbyists, but they are quite a different breed, for the most part, from the computer hobbyists of five years ago. I suspect that most of the circa 1975 hobbyists have either quit in disgust over the problems they encountered or else are alive and well, but working in the industry. The new hobbyists are less interested in designing circuits than in writing programs and finding out better ways to *use* microcomputers. They have become trapped by the enjoyment and mental expansion which computers bring. They are the new "hobbyists."

In line with this concept it seems appropriate to let *80* cover the world of the TRS-80 and *KM* the rest of the systems—all at a fundamental level that can help newcomers learn about computing.

In line with this basic concept we are looking for articles which will help newcomers over the hurdles. If you are a rank beginner, you might keep a log of the things that perplex you and then, when you have surmounted these problems, take the time to offer help to those who are to come after you.

Beginners need articles that explain in English about all of the mysteries of computers. They want to know about all the different kinds of printers and which they need to buy. They want to know about memory and storage devices. I have yet to see a good article anywhere on all of the different kinds of disk units. They want to know about I/O ports, about control systems, about languages and operating systems. They want up-to-date information. Get busy. We pay well for articles.

The Future

How can one look very far into the future of computers? The changes are coming on a monthly basis. It is almost all we can do to cope with the present, much less predict with success what things will be like in five, ten or twenty years.

Yet, when we look back on the past, we find that most of the things we have at present were reasonably predictable.

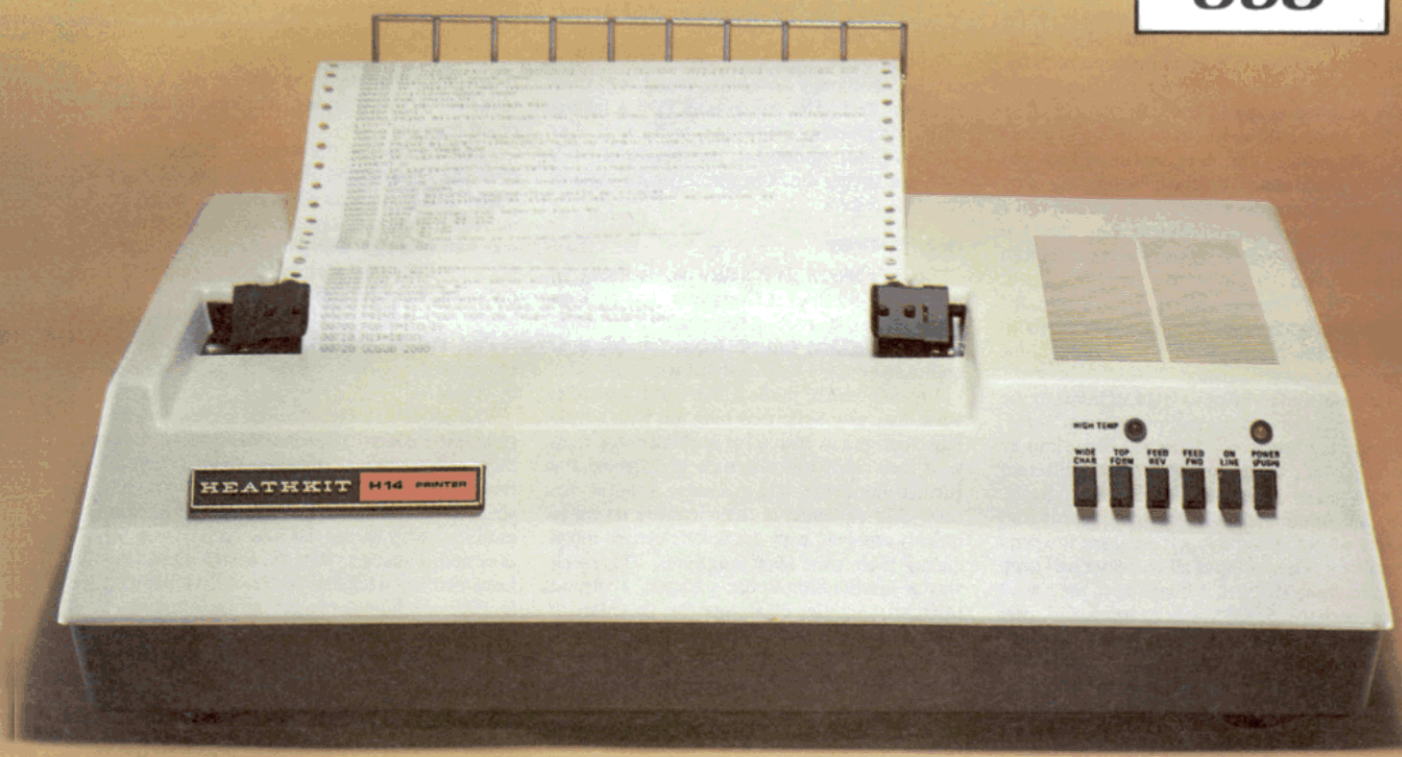
Microcomputers can save a whale of a lot of money and time (which is money) for businesses, so in the future, we are going to see them being used heavily. One of the more significant developments will be a universal electronic mail system. Once that is up and running I think micros will be getting into businesses at

Lew Kornfeld, the president of Radio Shack, attended the showing of the three new TRS-80 computer systems at the recent press conference in Ft. Worth. That's Lew on the left, me on the right.

The Model III was a good move, if not particularly newsworthy in view of the small changes between it and the Model I. It does pave a way toward eventually stopping production on the Model I, which FCC noise requirements would have dictated anyway.

Very sharp printout at very low payout

$595*



The Heath H-14 Printer gives you high-performance features at one of the lowest prices anywhere...

- 5 x 7 dot matrix and high quality impact printhead give you clear, easy-to-read images
- Standard 96-character ASCII set, UPPER and lower case
- Operator or software-selectable line widths; 132, 96, or 80 characters
- Compatibility with any computer having RS-232C or 20 mA current loop serial interface, with handshaking
- Sprocket paper feed, with adjustable spacing, to keep paper moving smoothly
- Sustainable print speed approximately 30 characters per second
- "Paper jammed" and "paper out" signals to prevent loss of data
- Selectable baud rates from 110 to 4800
- Convenience of standard fan-fold paper, 2.5 to 9.5 inches wide
- Chrome wire rack to keep paper neat

The H-14's remarkable price includes connecting cables, paper rack, paper, and ribbon – so you're all ready to run. And service for the H-14 is available at 61 Heathkit Electronic Centers in the U.S. or Canada.

Check out the microprocessor-based H-14 Printer today, in kit form or factory assembled. You'll find complete details in the newest, FREE Heathkit Catalog. Send for yours today, or pick one up at the nearest Heathkit Electronic Center. **Dealer inquiries on assembled units are invited, too.**

✓383

Complete service, so you're never left out in the cold

Heath®
Computer Systems

Send for FREE Catalog

Write to: Heath Company, Dept. 035-704. Benton Harbor, MI 49022

Visit your Heathkit Service Center

H-14 Printers are on display at the 61 Heathkit Electronic Centers throughout the U.S. and Canada. See your telephone white pages for the location nearest you.

In the U.S. Heathkit Electronic Centers are units of Veritechnology Electronics Corporation.

*In kit form, FOB Benton Harbor, MI. Also available completely assembled at $895.00, FOB Benton Harbor. Prices subject to change without notice.

CP-191

a much faster clip than they are at present. That use alone will more than pay for the computer.

The software business will, I expect, grow significantly for specific business uses. I'm still not wholly convinced of the place of the computer in the home, or even of the concept of the personal computer. I suspect that the media have been led astray by these terms.

The more I think about home applications for the computer, the more convinced I am that most of these applications will be taken over by dedicated microprocessors. Sure, we can run a microwave oven with our home computer, but with a lot less trouble we can build a chip into the oven to do the job. Ditto watering the lawn, security for the home, and ditto, I'm afraid, for almost every other application which comes to mind for the home. Sorry about that, home computerists.

As more and more of us work at home or use a computer for educational purposes at home, perhaps the home computer will come into its own. I don't think it will be much different from a school computer or an office computer. Oh, an automated index to our records and books would be handy. . .if we want to spend the time it takes to input all that data and keep it up to date. The idea is boggling to me since I have, perhaps, 5,000 books to index and, perhaps, a thousand records with God knows how many cuts on them. Then there are all those magazine articles I would like to be able to find. It comes down to whether or not I can afford a full-time secretary to put all that data into the computer so I can find the record or book I want. I can do that pretty well now, without the index—if I spend a bit of time searching. That may be more efficient than the index.

What do *you* think?

Threats from Japan

Virtually every Japanese electronics firm has had a microcomputer on the market. It was only a matter of time before some of these outfits started looking to see what they could do over here. One of the first units to come over was the Sord, but it was a quarter-hearted effort and unsupported by significant advertising, so nothing came of it.

At NCC in May I saw the first of NEC systems being shown. I am not yet sure that Nippon Electric will be coming over in force, but their success in Japan and their competitive edge against Apple may turn the trick.

More definite are the plans by Matsushita with their Panasonic and Quasar brand consumer electronics. They showed their Quasar system at the summer CES (June) in Chicago and generated much enthusiasm. I talked with them about this and they are projecting sales of about one million systems for 1981. At approximately $400, the computer is not much larger than those language translators, a hand-held unit. Even with all of its accessories, it will fit in a small attache case. It's ideal for the traveling businessman or salesman. They just might reach their goal in 1981, unless Radio Shack pushes hard with their very similar TRS-80 Pocket Computer.

Another interesting system shown at CES was from Casio. Casio says they intend to be the largest microcomputer firm in the U.S. by 1982. With enough software support and the well-known Casio advertising and marketing, they might do it.

Apple? Their sales are growing rapidly, but limited available software could be their Achilles' heel. The Apple may take a licking with competition from the NEC and new TRS-80 Color equipment, particularly if NEC makes a strong move to get software support for their system.

New Hardware

Tandy's Model III is a nice development, as is the TRS-80 Pocket Computer, but perhaps Radio Shack is reacting too much to the hardware competition and further splintering their ability to support their systems with software.

I remember the panic which came over Mits when they saw Sphere coming out with a 6800 based computer. Instead of pushing ahead hard with their 8080 based system and developing further hardware and software support for that, they squandered their lead by trying to quickly compete with the Sphere system introducing their own 6800 computer. The result was a multimillion-dollar disaster. I suspect their pushing the 6800 was the downfall of Mits, weakening their cash situation, curtailing their growth, and eventually forcing them to sell out to Pertec, where massive indifference to the micro market quickly sank whatever was left of Mits.

If Mits had pushed their advantage and not gone into a panic mode over the Sphere, which folded up as a result of poor design, an almost total lack of software support and insufficient financing, Mits might be one of the largest firms in the field today. Millions of dollars down the tubes.

I can understand the worry by Radio Shack over the Quasar and Panasonic pocket computers and it may turn out that the effort required to turn out Radio Shack's own system was well invested. But with six different computer models to support, even the resources of a billion dollar firm are straining beyond what seems practical.

The TRS-80 color unit looks good in response to the growing market share being taken by Apple.

The Model III TRS-80 is an appropriate response to the need of businesses and schools for more self-contained units and to the recent increase in Commodore sales resulting from their single unit system design. The software compatibility with earlier TRS-80 systems is a big plus and, I suspect, that the design considerations were not a big deal for Tandy. Model III is more a repackaging project, a successful one, I would say.

Come 1983, which manufacturer will be in the driver's seat?

INSIDE 80

by Ed Juge, director of computer merchandising, Tandy Radio Shack

Last month space did not permit telling you about our new printers in the Radio Shack line for 1981. Introduced at the same time as our new computers, they have been quite well received. I'd like to tell you about two of them now.

The first is our new TRS-80 Line Printer VI available September 30. It is a very low-profile, 14-inch wide, 132-column printer, which can be pinch-fed, but comes with a removable, adjustable tractor feed. It has a nine-wire dot-matrix head, and produces four print sizes (5, 7.5, 10, or 15-characters-per-inch), plus graphics characters. Of course it offers upper and lowercase. Its speed is 100 CPS, and the LP VI is bi-directional. The average throughput is 33-lines-per-minute and it uses our standard parallel port interface.

Versatile Feed

The versatile feed system allows you to use tractor-fed forms from four to 14-7/8-inches wide, or even single sheets of paper (pinch-feed). It's also rated for the original and two copies. Its overall size is only 6⅓ × 13⅓ inches, and it weighs only 28 pounds. Exclusive of the cable, it's only $1,160.

The other new printer is our TRS-80 Plotter/Printer, scheduled in limited quantities, for the end of November. If your primary requirement is plotting, but you don't want to buy a separate printer to list your programs, this might just handle both of your needs. This unique intelligent plotter draws with a standard ballpoint pen, on a continous roll of pin-fed, 9-inch wide paper. It can handle complex plots and graphs with outstanding resolution, and it can print upper and lowercase letters, approximately nine per inch, or 75 per line, at a nominal five-lines-per-minute.

The TRS-80 Plotter/Printer uses our parallel interface, weighs in at 26 pounds, and is 7½ × 18¾ × 14-4/5 inches. The price, excluding cable is $1,460.

This might be a good time to publicly reply to a fairly common question. . .and, I admit, some complaints. The complaints seem to revolve around a customer who bought a TRS-80 and Scripsit, but wanted someone else's printer. Now he can't get them to work together properly. Nor can he get anybody at Radio Shack to tell him how to make them work. work.

Well, I can certainly sympathize. At one time, I did a bit of drooling over one of the non-Radio Shack word processing printers. I was

Inside to page 40

WE HAVE A FULL HOUSE

DON'T GAMBLE

Buy Only From a "Factory Authorized Source"

THE STOCKING SOURCE

PRINTERS

	List Price	Your Cost
Okidata Microline 80	$ 945.	Ask for Our Price $800
Anadex Model DP-8000 or DP-8000AP	$1095.	Ask for Our Price $895
Anadex Model DP-9500 or DP-9501	$1650.	Ask for Our Price
Epson Model TX-80B Friction Feed	$ 710.	Ask for Our Price
Epson Model TX-80B Tractor Feed & Graftrax	$ 799.	Ask for Our Price
Epson Model MX-80	$ 699.	Ask for Our Price

INTERFACES

Okidata Microline 80 Tractor Feed . . $100.

Okidata Microline 80 RS-232 Interface with 256 Character Buffer . . . . . $200.

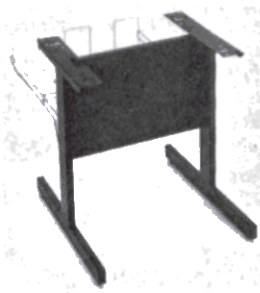
All above Printers — Cable from Printer to TRS-80 . . . . $ 35.

Epson-Serial Interface & Cable . . . . . $ 90.

Epson-IEEE 488 Interface & Cable $ 80.

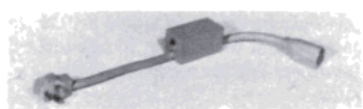
Epson-Apple Plug-in Interface & Cable . . . . . $110.

PRINTER STANDS



Systems Furniture Universal Printer Stand without top, but with paper basket . . . . . $120.

TRANSIENT CLIPPERS



The ideal Line Voltage Transient Clipper from PDF protects against • High Energy Voltage Transients • On-Off Switching • Lightning Induced Transients. Model C-120Q (Other Models Stocked) $57.50

ASK FOR OUR INSTANT DISCOUNT!

CALL TOLL FREE
U.S.A.
1-800-521-2764
MICHIGAN
1-800-482-8393

WRITE TO: ↙438
"The Stocking Source"
23995 Freeway Park Dr.
Farmington Hills, MI
48024

↙Reader Service—see page 226

80 INPUT

"The next time you...use the VTR to record that great movie from TV... label yourself...a thief, or worse, a pirate."

Computronics' Rebuttal

I wish to take issue with the statement made by Peter J. Brennan on Page 11 of the July, 1980 issue of *80 Microcomputing*. In that issue, Mr. Brennan states tht Computronics is "too expensive." I wish to point out some benefits one receives when subscribing to the *H & E Computronics Newsmagazine* for $24 per year.

1. Twelve monthly issues of the magazine are devoted largely to the serious TRS-80 owners rather than the hobbyist.
2. A 48-page catalog containing 180 software items written by established houses.
3. A money-back guarantee on all the software we sell (and even a money-back guarantee on disk drives and printers).
4. A free cassette containing five programs.
5. We have a special help line, and in-house programmers who will answer any question related to the TRS-80, including questions about hardware or software, wherever purchased.

In closing I wish to quote a part of a letter that we published in our magazine.

"...I very much appreciate your software refund policy. I was astonished at the speed with which you refunded me the price of the Income Tax Pax B program that I found not very useful at all. You may be sure that when I buy software in the future, you will be my first source. I don't want to return it...I want it to work right...but it's nice to know I am not stuck with useless stuff."

Who gave us such a nice compliment? Peter J. Brennan, the writer of the negative remark published in your July, 1980 issue.

Howard Y. Gosman, Publisher
H & E Computronics, Inc.
Spring Valley, NY

A Hobbyist's View

As a would-be computer hobbyist, and subscriber to several magazines, I have difficulty with the recent articles on theft and pirating of programs.

Since no one seems to write in behalf of the hobbyist, I'll take a shot at stating an alternate view. As usual, the press jointly pursue an issue and wholesale articles are written expressing a position on a subject affecting the publishing world directly.

My main objection is the association of theft and/or pirating as randomly applied to anyone accepting a program other than by direct purchase. It is bad enough to label and name call, but to offer rewards for arrest and conviction of these people is really getting small. Worse still, it will not cure what you believe to be a problem. The truth really is simple...because programmers know very well that programs are hard to protect and therefore control, prices are set high enough to recover a dollar percentage which will return a viable profit on investment.

Hobbyists in computing, as in radio, fishing, music and others, form common groups to exchange ideas, and yes, to exchange material or to jointly purchase items or materials. To call the members of such groups thieves or pirates would be ridiculous.

Consider the following example of a similar industry and problem.

Recording equipment is sold in various forms at most major stores in any community. These devices record either audio, electrical impulse, video or combinations of these. The next time you or a friend use the VTR to record that great movie from TV, remember the $25 million cost to produce it, the $400 daily rental loss, the $4.50 ticket price not paid to the local theater, and in keeping with your thinking, label yourself and your friend a thief, or worse, a pirate.

I personally have 30 to 40 hours of taped movies; great movies, but I didn't pay for them. More importantly, I'm not selling them. That is my point; by my way of thinking, no one is a thief or pirate unless they take something from someone and use that something for personal gain. You have not proved to my satisfaction that this is, in fact, being done with computer programs. Rather than inhibit growth, users groups and exchanges of information have generated demands that have pushed home computer technology ahead at a faster rate than originally dreamed was possible.

Ronald Dudeck
Ontario, CA

A Better Byte Loader

The RSM-2 machine language TRS-80 monitor from Small Systems Software, is a high quality product. It can do much more than Radio Shack's T-BUG and should be considered a necessity for any serious 80 user.

The 16K cassette version sells for $26.95 and a symbolic listing is available for $7.50 to those who have purchased the tape.

I have a modification to the RSM-2 which improves its byte loading utility.

The byte loader is activated by the U command of the RSM-2. To enter the loader program, type a U and follow it with the hexadecimal destination address of the first byte to be loaded. Hit the ENTER key and you will see the address displayed on the video monitor. You can then enter any desired bytes in hex code. Each character will appear on the monitor as you type, with a format consisting of an address followed by eight bytes per line, much like that of the familiar DUMP command.

While using the byte loader, three commands are available and can be used at any time following the entry of a complete byte. Pressing L displays the next destination address and starts a new line. The left arrow deletes the last byte entered. BREAK exits from the loader and returns you to the RSM-2 command mode.

Chesney E. Twombly
Kennebunk, ME

Twombly subroutine A.

This symbolic listing shows the changes to RSM-2 that are required in order to implement the TRS-80 better byte loader. RSM-2 is the 16K cassette version of the TRS-80 monitor available from:

```
                        ADRO    EQU     6E2B
                        KEYB    EQU     6D25
                        CRT     EQU     6FFE
                        SP1     EQU     6E39
                        USER    EQU     6F00

                                ORG     6C18

6C18  3E C3                     LD      A,C3
6C1A  32 00 7F                  LD      (USER),A
6C1D  3E 01                     LD      A,1
6C1F  32 01 7F                  LD      (USER+1),A
6C22  3E 7E                     LD      A,7E
6C24  32 02 7F                  LD      (USER+2),A
6C27  C9                        RET

                                ORG     6EB1

6EB1  00                        NOP
6EB2  00                        NOP
6EB3  00                        NOP

                                ORG     6EB0

6EB0  CD 18 6C                  CALL    6C18
6EC0  00                        NOP
6EC1  00                        NOP
6EC2  00                        NOP

                                ORG     7E01

7E5C  07                BYTE    RLCA
7E5D  07                        RLCA
7E5E  07                        RLCA
7E5F  07                        RLCA
7E60  C5                        PUSH    BC
7E61  47                        LD      B,A
7E62  CD 3D 7E                  CALL    INHEX
7E65  80                        ADD     A,B
7E66  C1                        POP     BC
7E67  C9                        RET

                                END
```

Twombly routines cont.

Browning Attacks

I keep reading of the outrage and frustration that some folks suffer or seem to suffer from having their programs copied by scoundrels who won't buy them.

Let me tell you that your outrage is small compared to that of my own. I am a user of programs, not a writer, since I am much too busy and too inexperienced to write the complex and badly needed programs to assist me in the operation of a fairly large insurance agency.

I have purchased many programs in an attempt to get what I need to run $11,000 worth of computers, and have begun to realize considerable contempt for sellers of programs which are advertised to do something great. The ads do not tell what they don't do; sometimes they don't even work at all, and sometimes they disable the main program and make it unusable.

Consider a program called Pencil/Pal, supposed to add names from a mail list to a form letter created by Electric Pencil. It works... sort of... but the ad doesn't say that it disables the most valuable feature of Electric Pencil, that of being able to handle words in a continuous string, with Pencil dong the work of justifying and placing everything where it belongs. Also lost is the variable line length: You are stuck with 62 characters to a line and the necessity of a carriage return at the end of every line.

The $35 is gone and the seller of the program won't give it back even though his program creates more problems than it solves.

I can give you many more examples since I have about $500 worth of programs that don't work. I've got some darn good ones, too, but I've had to go through pure hell to find them.

You won't find me in the sympathetic group for those who cry about having their programs ripped off and not getting paid for what they supposedly do such a good job of. I've been "had" too many times by these guys. The ones who really do a good job get my money and my appreciation, and more than that, they get referrals.

I am glad to part with my bucks to the guy who knows that he has to do a good job and take a little risk and sometimes even do it over to get it right.

I think the one who does a poor job deserves some "advertising" too.

One other thought before I sign off... I sure get tired of waiting for that "fantastic new program." You know the one. Full color ads, all those features. But after you send your $100, they tell you it won't be available for 60 to 90 days. Are they testing the market to see if there is a demand for the item, and if they get enough positive response they try to invent the product??

Goodnight VTOS wherever you are.

Kaye Browning
Roy, UT

MicroComputer Responds

As author of Pencil/Pal, I would like to respond to Kaye Browning's criticism of our low cost ($35) form letter generating program. Pencil/Pal allows the user to automatically generate form letters from address and letter files which are created using the Electric Pencil, Scripsit, or any other text editor or BASIC program that produces ASCII files. Letters may be printed to a subset of the address file by specifying one or two "select codes." The select code may be any string of characters in the address field or the user's own custom code (account #, phone #, amount due, etc.).

In contrast to Ms. Browning's comments, Pencil/Pal performs exactly according to its extensive documentation. Her comment that Pencil/Pal has "disabled the most valuable feature of Electric Pencil" is misleading. In fact, Pencil/Pal does not tamper with Electric Pencil code at all. I believe she is referring to the fact that the user must end each line of the form letter with a carriage return (ENTER). This feature allows the user to format the output exactly as desired (one is not "stuck with 62 characters to a line"). The user may elect to right-justify or hyphenate his letter manually, a small inconvenience which may yield a more professional looking output than is possible with many word processors (hyphenation reduces the disturbing gaps between words that often occurs when the Electric Pencil justifies).

I believe that Pencil/Pal is a functional, bargain-priced software package that saves a considerable amount of labor for those individuals or small businesses that require automatic generation of form letters.

I would also urge anyone that is watching their hard earned software dollars to request a copy of a program's manual before investing in the actual software. Advertising copy cannot always tell you all the features or limitations of a program. The manual for Pencil/Pal is available for $5 (applied toward purchase price of $35).

Rodney B. Murray, Ph.D.
President, MicroComputer Specialists
Elkins Park, PA

Twombly subroutine B.

cont. from previous page

```
7E01  EB          START  EX    DE,HL
7E02  CD 2B 6E    ADDR   CALL  ADRO
7E05  CD 25 6D    BYTES  CALL  KEYB
7E08  CD FE 6F           CALL  CRT
7E0B  FE 1D              CP    1D
7E0D  C2 1A 7E           JP    NZ,CKAD
7E10  CD FE 6F           CALL  CRT
7E13  CD FE 6F           CALL  CRT
7E16  2B                 DEC   HL
7E17  C3 05 7E           JP    BYTES
7E1A  FE 4C       CKAD   CP    4C
7E1C  C2 22 7E           JP    NZ,LOAD
7E1F  CD 02 7E           CALL  ADDR
7E22  CD 44 7E    LOAD   CALL  CONVT
7E25  CD 5C 7E           CALL  BYTE
7E28  77                 LD    (HL),A
7E29  CD 39 6E           CALL  SP1
7E2C  23                 INC   HL
7E2D  7D                 LD    A,L
7E2E  E6 07              AND   7
7E30  FE 00              CP    0
7E32  CA 02 7E           JP    Z,ADDR
7E35  C3 05 7E           JP    BYTES

                         ORG   7E3D

7E3D  A7          INHEX  AND   A
7E3E  CD 25 6D           CALL  KEYB
7E41  CD FE 6F           CALL  CRT
7E44  D6 47       CONVT  SUB   47
7E46  F2 3D 7E           JP    P,INHEX
7E49  C6 06              ADD   A,6
7E4B  F2 53 7E           JP    P,OK
7E4E  C6 07              ADD   A,7
7E50  F2 3D 7E           JP    P,INHEX
7E53  C6 0A       OK     ADD   A,0A
7E55  FA 3D 7E           JP    M,INHEX
7E58  C9                 RET

7E59  CD 3D 7E    INBYTE CALL  INHEX
```

80 AID

File Transfer Aid

Ed Maurer (July 1980) asked about communication between the TRS-80 and Digital PDP 11/70. I do it all the time between the house and the office where we run a DEC PDP 11/40 using the RSTS/E operating system and DEC's PIP program.

Ed doesn't say what operating system his machine uses but almost all DEC monitors have some version of PIP and would probably work. Of course the DEC machine will have to have a dial-up keyboard port.

At home I have a 48K system with a Tandy RS-232 board, a Tandy Telephone Interface modem and a couple of disks. The key is the software. I run Lance Micklus' ST80-II Smart Terminal program.

Woods Martin
Houston, TX

Power Outs

All microcomputer owners have had to live with disruptions caused by power failures. For reasons unknown to residents of Ashland, Oregon, we seem to experience this problem at least once (if not more times) each month.

Hardware is available to protect the CPU from power surges, but is anything available that could supply back-up power during outages—so I wouldn't lose the program in the memory?

J. Ngan
P.O. Box 621
Ashland, OR 97520

Try May Day power supply devices from Sun Research, Inc.—Eds.

80 DEBUg

SET/RESET Bug

The following is a correction to the SET/RESET subroutine call which appeared in my article in the February, 1980 issue. The key is the loading of the HL register pair with the address of a right parenthesis. Unfortunately, I overlooked this omission when proofreading the article.

```
        LD      HL,RETRN
        PUSH    HL              ;put return address on
                                stack
*****   LD      HL,18DH         ;point to a right paren-
                                thesis
                                *****
        LD      A,set/reset code
        .
        .
        etc.
```

I have received numerous letters and phone calls about the article. There seems to be quite a large number of your readers who are interested in assembly language programming. Regrettably, I simply do not have the time to answer all the letters.

Wes Thielke
Mercerville, NJ

Revised Line Formatter

In my subroutine for formatting lines (page 162 of the July 1980 issue) there is a bug lurking which can at times cause an illegal function in line 30040. The malfunction is caused by the addition of 2 in line 30000 to set the number of lines to be printed. If the second line is not needed, a negative number will be sent to the argument of the RIGHT$ instruction in line 30040, and this generates the FC message.

Alterations to fix the problem simply made the routine more bulky and complex, so I decided to revamp the whole thing. The revised listing follows:

```
5     CLEAR 510
10    REM * PUT LINE TO BE PRINTED INTO A$
20    GOSUB 30000
30    END
30000 R3=0:A$=A$+CHR$(129):R4=LEN(A$)
30010 FOR R1=41 TO 1 STEP -1:R$=MID$(A$,R1,
      1):IF R$=CHR$(129)
      THEN 30040
30020 IF R$=" " THEN 30030 ELSE NEXT
30030 R3=R3+1:R1$(R3)=LEFT$(A$,R1-1):A$=
      RIGHT$(A$,R4-R1):R4=R4-1:GOTO 30010
30040 FOR R2=1 TO R3:LPRINT R1$(R2):NEXT
30050 R5=LEN(A$):LPRINT LEFT$(A$,R5-1):RE-
      TURN
```

In response to inquiries, this routine may also be used with printers having different print widths. Resetting the value of R1 in line 30010 to the proper width plus one is the only change necessary. For example, for a print width of 80 characters change the line to read:

```
30010 FOR R1=81 TO 1 STEP -1:. . . etc. . .
```

John D. Adams
Sylmar, CA

Swords and Sorcery Fix

All right, all right. So nobody's perfect. Granted, there were problems with our listing of the Swords and Sorcery program in the August issue. But it wasn't all our fault. You guys who insist on writing those 255 character program lines have to share some of the blame.

Since our print driver routine sources the program from an ASCII formatted file, super-long lines just don't make it. So, from now on, keep those lines around 240 characters and don't forget to include the line numbers when you're counting.

Listed below are the corrections you've been waiting for.—Eds.

```
10 DATA 160,190,191,180,184,191,191,191,191,191,188,144
   ,160,186,191,191,191,191,191,191,191,191,191,191,1
   91,191,188,188,144,160,190,191,147,175,191,191,191
   ,191,191,191,191,191,159,131,179,191,181,160,190,1
   91,191,151,160,191,191,168,191,186,
15 CLS:
   CLEAR 250:DEFINT L:XX=458:GOSUB 65:FOR X=1TO9:READ
    Y,Z:A(X)=Y:B(X)=Z:NEXT:DATA 15898,3,15961,7,16023
   ,16,16086,17,15969,4,15907,2,15844,1,15781,0,15717
```

Swords and Sorcery fix conts.

Program continues

A Second Opinion

Mr. Brennan has indicated in his letter in the July issue that there are problems with the Scripsit program, and that as far as he is concerned, Electric Pencil is a much better word processor. He is correct in the former, and probably correct—for him—in the latter, but I suspect the fact that he had EP first has somewhat biased his opinion. After all, nobody is denying that EP is an excellent program.

Here is what I did to overcome the problem that both he and I (and no doubt others) had when using the program with a serial printer. The specific problem I had is related to the way my printer (an Anderson Jacobson 841 I/O) handles combinations of carriage returns and line feeds. The printer ignores the first line feed after a carriage return. Unfortunately, Scripsit generates a line feed after a carriage return whenever it wants to do double line spacing or to insert extra lines between paragraphs (curse it!).

As Mr. Brenner states, if you tell the program that you want triple-line spacing when you really want double, you get it, but your page divisions are thrown out of place. Consequently, a little more devious method has to be used.

After a little research with a disassembler and a character search routine (both in MON4, by Hubert S. Howe), I used Apparat's excellent program Superzap to make the following changes to Scripsit.

Memory location 7171H—(disk location 0B1F1H) and memory location 7185H—(disk location 0B205H)—change value 0AH (line feed) to 0DH (carriage return) and the line feed problem vanishes.

The most obvious criticism of Scripsit concerns the instruction tapes, or rather the fact that no other way of learning Scripsit is offered. The tapes are in fact quite good, but I and quite a lot of other people who use the program have met word processors before. The idea of spending three to five hours listening to the tapes nearly stopped my wife from using the program altogether. Why there isn't an alternative way of discovering how to use the program, Tandy only knows.

Once you have toiled through the learning process, the documentation is perfectly adequate for reference purposes. However, I have some objections to some of the things I cannot do with text formatting statements. Despite all claims to the contrary, it is not possible to display on the video the text as it will be printed, since the video display has no provision to do formatting according to the imbedded printer formatting instructions. The second problem is the difficulty of imbedding control characters (such as back space) into the text. The third is that I cannot find a way to do "reverse paragraph indenting," i.e., having every line but the first of a section be indented by say, eight characters, while retaining right margin justification. This last may not be a normal requirement, but it is very useful if you deal with numbered sections in technical documents.

EXATRON STRINGY FLOPPY T.M.

SPEED
LOW COST
RELIABILITY

Exatron is a California based corporation that has been in business since 1974. As well as the Stringy Floppy, Exatron designs, manufactures and sells state-of-the-art electro-mechanical equipment for a variety of commercial and industrial applications. Exatron is an established supplier of automatic test equipment to manufacturers, and large OEM users, of integrated circuits worldwide.

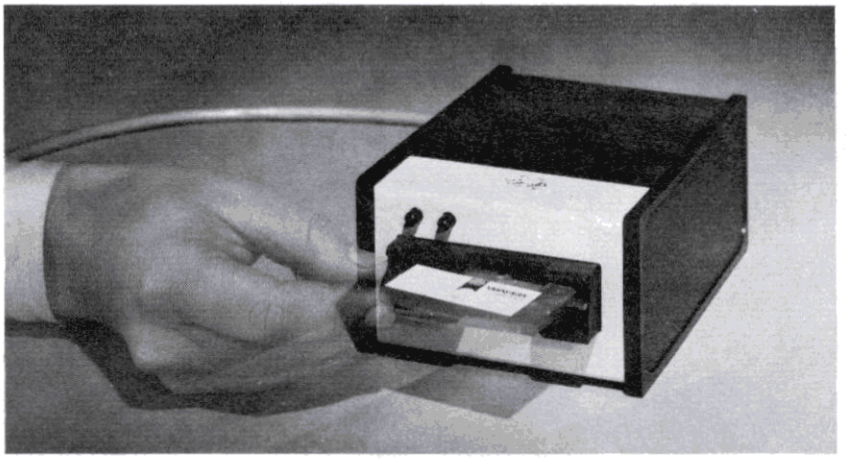
The software in every ESF adds a parity bit to every byte saved on tape, and a checksum to the end of every file. These are checked both after recording data and upon replay, any detected error is indicated by a message on the video display. This system of automatic error checking gives confidence in any data saved, also each wafer is rated for at least 2,000 complete passes past the record/replay head.

- ►Assembled and tested
- ►All operating software in ROM
- ►Fully automatic operation
- ►Professional quality
- ►No Expansion Interface required
- ►Large Owners Association
- ►High speed operation
- ►Extremely reliable
- ►No technical knowledge needed

WHAT IS IT?

The Exatron Stringy Floppy (ESF) is an extremely fast, reliable, economical alternative to cassette or floppy disk storage of computer programs or data.

Totally self-contained, the ESF has no buttons, switches, knobs or levers to adjust or forget. All of ESF's operations are under the computer's control.



TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.

HOW DOES IT WORK?

The ESF uses a miniature tape cartridge (called a 'wafer') as the data storage medium, about the size of a business card and 3/16th of an inch thick. The tape used inside the wafer is a special Mylar based Chrome Dioxide type, specially developed for digital applications. Wafers are available in several lengths, 5 feet being the smallest and capable of holding up to 4 thousand bytes of information – the 75-foot wafer is the largest available and can hold up to 64 thousand bytes of data.

The wafers contain a single reel of the special tape connected as a continuous loop, the ends being spliced together with a piece of reflective tape. In operation the ESF drive unit pulls the tape from the center of the reel inside the wafer, causing the entire reel to rotate. Thus, the tape automatically winds itself around the outside of the reel at the same rate as which it is pulled from the center. This process is similar to that found in an 8-track cartridge.

The ESF transport mechanism is very simple, consisting of a precision die-cast aluminum block – with a capstan, drive motor and magnetic record/replay head mounted on it. The wafer loads into a slot in the casting (it will only fit the correct way) and the tape is driven at a single point by the capstan, past the record/replay head.

HOW DO YOU USE IT?

Once connected to your computer the ESF operating system needs to be activated—simple. Just type 'SYSTEM'(enter), and in response to the ? prompt type '/12345' (enter). Your TRS-80 will instantly display the ESF sign on message 'EXATRON STRINGY FLOPPY VERSION 4.1', and from this point onwards you will have the extra commands '@LOAD', '@SAVE' and '@NEW' recognized by your TRS-80.

The ESF's operating system is built into the electronics of the unit, in much the same way that BASIC is built into the computer, so it is always available – the SYSTEM command is to let your computer know that the ESF has been connected. If you normally reserve some memory for subroutines then the ESF software will relocate itself under your selected top of memory. The ESF uses only 4 bytes of your available RAM, these bytes are used to 'point' to the 2048 bytes of software in the ESF unit itself.

WHAT'S THE CATCH?

Well, the only catch that most people find is that they have to actually pay Exatron for their unit! Even this is no big deal.

Starter Kits are available with the Exatron Stringy Floppy, a supply of wafers, a bus extender and a selection of useful programs – for $299.50.

Through regular advertisements in both Kilobaud Microcomputing and 80 Microcomputing, owners are kept informed of the latest developments in wafer-based software. Plus hundreds of user 'workshops' all over the country, so you can always be sure of being near to another ESF owner.

Exatron also gives a 30-day full money-back guarantee, with a 1 year parts and labor warranty on the unit.

If you have any questions about the ESF then give Exatron a call on the Hot Line (outside CA) **800-538-8559.**

East Coast customers can call **800-343-4424** (inside MA **617-899-3862**)

Open House Workshops take place from 9 am till 1 pm every Saturday at Exatron's factory in Santa Clara, and on the East Coast the last Saturday in each month at Micro Communications, 80 Bacon Street, Waltham MA 02154. All are welcome.

exatron
181 Commercial St.
Sunnyvale, Ca 94086
408-737-7111

✓3

✓Reader Service—see page 226

```
      ,0:R$=" OUR BUNGLING HERO "
 20 RANDOM:PA=2:B$(1)="CLANK ":B$(2)="SLASH ":B$(3)="WOO
     SH ":B$(4)="BONG ":B$(5)="CRASH ":B$(6)="BING ":B$
     (7)="CLANK ":S$(1)=CHR$(160)+CHR$(183)+CHR$(181)+C
     HR$(183)+CHR$(181)+STRING$(60,32)+STRING$(4,149):Q
     $=CHR$(149):K$=CHR$(132)

120 IF RND(PN)*2<=RND(PN)*RND(2) GOSUB65:PRINT"A DRYAD
     HAS OFFERED TO BE YOUR":PRINTTAB(12);"GUIDE":PRINT
     TAB(6);" DO YOU WISH IT ?":GOSUB110:GOSUB90:IF AN=
     89,F=1 ELSE IF RND(0)>.2 GOSUB65:GOSUB 500:GOSUB10
     5

140 PRINT"DO YOU WISH TO CONSULT THE GREAT";:PRINTTAB(1
     2);"ORACLE ?":GOSUB110:IF AN=78,180 ELSE CLS:XX=20
     2:GOSUB 65:PRINT:PRINT"AHA! TO GAIN FAVOR WITH THE
      FAT  ONE AND GET THE POOP YOU NEED    YOU MUST FI
     RST APPEASE HIM.":PRINT

540 PRINT"YOU MUST STOP AND REST BEFORE GOING ON.":IF F
      PRINT"THE NYMPH THINKS THAT THE DUNGEON IS LESS T
     HAN";ABS(L-20);"YERBS AWAY":ELSE PRINT"YOU HAVE TR
     AVELED";INT(ABS(DT-L)*.75);"FARBBLE WARFERS"

570 PRINT:PRINT"YOU HAVE BEEN CAPTURED BY GOBLINS":IF E
     <>1,600 ELSE PRINT"THEY WANT THE SWORD THAT ONCE B
     ELONGED TO THE OLD ONES -":PRINT"WILL YOU TRADE IT
      FOR YOUR FREEDOM ?":GOSUB110:IF AN=78,600 ELSE E=
     -.8:PRINT"IT IS THEN AGREED":GOSUB1

600 Q=RND(30):IFG>=Q PRINT"THE GOBLIN LORD FREES YOU FO
     R";Q;"GOLD COINS":G=G-Q:GOTO580:ELSE IF W<=0,R4=8:
     PRINT"YOU ARE ENSLAVED":GOTO3000:ELSE PRINT"YOU AR
     E SOLD TO THE SATYRS BY THE GOBLINS":GOSUB760:GOTO
     580

610 IF R=0 PRINT:PRINT"LOOK! THERE IS THE ENTRANCE TO T
    HE DUNGEON":GOSUB105:PRINT"H";:FOR EX=1TO61:PRINT"
    M";:GOSUB80:NEXT:PRINT" !":PRINT"THERE APPEARS TO
    BE A GUARD":GOSUB105:PRINT"IT'S TOO DARK TO SEE FR
    OM HERE - MUST GET CLOSER ":GOSUB10

680 VA=448:VB=462:FOR V3=1TO2:FOR LZ=VA TO VB:PRINT@LZ,
     E$(WX);:GOSUB90:PRINT@LZ,E$(8);:WX=(3-(WX-5))+5:NE
     XT:GOSUB 660:VA=462:VB=476:NEXT V3:PRINT@VB,E$(4);
     :GOSUB685:PRINT@VB,E$(5);:GOSUB685:PRINT@VB,E$(5);
     :PRINT@VB,E$(4);:FOR X=1TO3

682 PRINT@492,E1$:GOSUB100

690 PRINT@490,E3$;:GOSUB100:PRINT@490,STRING$(8," ");:G
     OSUB90:PRINT@490,E3$;:GOSUB100:PRINT@490,E4$;:GOSU
     B100:PRINT@VB,E$(2);:GOSUB100:CLS:ON RND(4)GOSUB70
     0,960,960,700:R=1:W=W+1

691 PRINT:PRINT"OK,YOU'VE FOUND THE PRINCESS":PRINT"LET
     'S GET OUT OF HERE !":GOTO620

780 PRINT"WILL YOU AGREE TO THESE TERMS ?":GOSUB110:IF
    AN=78 PRINT"OH DID YOU MAKE THEM MAD - THEY DO YOU
     IN AND TAKE THE WOMEN":R4=6:GOSUB100:GOTO3000:ELS
    E PRINT"THEY TAKE THE WOMEM":IF RND(0)>.03 PRINT"T
    HEY CURSE YOU":K=-5

860 CLS:SP=540:FOR X3=1TO RND(3):X6=1:GOSUB885:FOR X4=1
     TORND(50)*10:NEXT X4:CLS:GOSUB85:NEXTX3:X6=3:GOSUB
```

Swords and Sorcery fix conts.

I must admit that I have not been troubled by Mr. Brenner's problem of not being able to read the disk directory directly from Scripsit, but then I didn't really expect to. I am annoyed by EP's habit of putting its own file extension on things—I find that Scripsit is an excellent means for generating source files for the disk assembler (*much* better than the editor which comes with that assembler), and for that as well as for sections of a long document, I want to add my file extensions.

I don't find that the command formats of Scripsit are particularly cumbersome, particularly since they allow you to have several blocks marked at any one time—but that is one of those things that is subjective. However, the fact that EP loses characters at the ends of lines is a serious defect for anyone who types quickly (not me, I hasten to add).

For anyone considering buying either of these programs, I can only hope that these debates in the letter columns are a help. In my opinion, either is a good buy; Scripsit is $50 cheaper, and I prefer it. Admittedly, it won't do some of the exotic format control which is available under UNIX on a PDP-11/70—but for a few hundred thousand dollars less, it seems pretty good value for the money.

R. J. Lighton
Wood-Ridge, NJ

Subs for INKEY$

I was reading Mr. Martinott's letter in the July issue which concerned Mr. Himler's article in the April issue, when I had an idea. I tried it and it works! I submit the following two subroutines:

```
1000 W$=""
1010 W$=W$+INKEY$:IF LEN(W$)<NC% THEN 1010
     ELSE RETURN


1500 W$=""
1510 W$=W$+INKEY$:IF RIGHT$(W$,1)<>TC$ THEN
     1510
1520 W$=LEFT$(W$,LEN(W$)-1):RETURN
```

Subroutine 1000 will return with a string of length NC%. Subroutine 1500 will return with the characters input preceding the character in TC$. Neither of these subroutines prints the string being input. If that is required PRINT@ PA%,W$;: could be inserted between INKEY$: and IF in lines 1010 and 1510.

Note that these subroutines have fewer restrictions than the regular input statement. Subroutine 1000 will input anything that can be input with INKEY$, which includes all keyboard inputs except BREAK.

Subroutine 1500 also excludes the termination character in TC$. In addition to the control codes that can be directly keyed in from the keyboard, such as line feed (the down arrow) or carriage return (ENTER), the ASCII control codes decimal 2 through 26 can be input by pressing the shift, down arrow, and a letter key B through Z. When you press the shift and down arrow, you get code 26 and then the control code when the letter key is pressed while holding down the shift and down arrow. This

```
    885:GOSUB100:X6=1:GOSUB885:GOSUB100:X6=4:GOSUB885:
    GOSUB105:CLS:GOSUB85:X6=2:GOSUB885:GOSUB95:CLS:X6=
    2:GOSUB885:PRINT@287,Q$
862 PRINT@351,K$:GOSUB90

870 PRINT@660,"A ROPE HAS BEEN LOWERED":X6=1:GOSUB885:G
    OSUB105:PT=0:Y=RND(4):PRINT@724,"YOU HAVE BEEN RES
    QUED BY ":IF Y=1 GOSUB 700 ELSE IF Y=2 PRINT"OH NO
     !":GOSUB570 ELSE IF F PRINT"THE NYMPH" ELSE PRINT
    "AN OLD LADY":W=W+1

1000 IF X=2 PRINT"HE THRUST HIS SWORD STRAIGHT FOR THE
    BODY !":GOTO1030:ELSE IF X=3 PRINT"HE ATTEMPS TO S
    EVER YOUR HEAD IN A SINGLE BLOW !":GOTO1030:ELSE I
    F X=4 PRINT"HE TWIRLS THE MACE DIRECTLY TOWARD YOU
    R HEAD !":GOTO1030

1010 IF X=5 PRINT"HE SWINGS HIS MACE SAVAGELY AT YOUR B
    ODY !":GOTO1030:ELSE IF X=6 PRINT"HE GLANCES YOUR
    BLOW AND LAYS ON WITH HIS SWORD !":GOTO1030

1015 PRINT"HE KICKS SAND IN YOUR FACE AND SWINGS HIS SW
    ORD TO CLEAVE THE   AIR AND YOUR HEAD ALONG WITH I
    T"

1030 IF RND(0)<=.5+.3*H2/W2,1050 ELSE PRINT"YOU'RE HIT
    !":H1=H1-.2:H2=H2-.2:GOSUB100:PRINTTAB(15);"OOOOF
    !!":GOSUB95:IF H1>=.05 PRINTTAB(30);"YOU STAGGER A
    WAY . . . . . .":GOTO980:ELSE PRINTTAB(30);"YOU'RE
     DOWN !!!":GOSUB100

1050 X=RND(6):IF X=1 PRINT"YOU STOP HIS BLOW WITH YOUR
    SWORD AND BACK AWAY !!":GOTO1085:ELSE IF X=2 PRINT
    "YOU DUCK UNDER HIS SWORD - VEER FROM HIS MACE AND
     ATTACK !":GOTO1070:ELSE IF X=3 PRINT"YOU PARRY TH
    EN ATTACK !":GOTO1070

1060 IF X=4 PRINT"YOU KICK HIM IN THE SHINS AND SCAMPER
     AWAY !":GOTO 1095:ELSE IF X=5 PRINT"YOU STOMP HIS
     TOES WITH YOUR BOOT !":GOTO1095:ELSE PRINT"YOU SL
    ASH LEFT !";:IF RND(3)=1 PRINT:ELSE PRINT"YOU SLAS
    H RIGHT !"

1070 FORX3=1TOH3:IF RND(0)<=.1 PRINT"YOU MISSED HIM !!!
    !":ELSE X=RND(H3):IF X=1 PRINT"YOU GOT HIS LEG !":
    W2=W2-(DS+H2/5):W3=W3-(DS+H2/5):ELSE IF X=2 PRINT"
    YOU'VE SLASHED HIS ARM":W2=W2-(DS+H2/3):W3=W3-(DS+
    H2/5)

2120 PRINT@347,S$(1):GOSUB90:PRINT@347,"      ";:PRINT@4
    12,S$(2);:GOSUB90:PRINT@412,S$(1):PRINT@604,"SLURP
     !":GOSUB90:PRINT@663,"BU";:FOR X=1TO10:PRINT"R";:
    NEXT:PRINT"P !!";:GOSUB100:PRINT"   HIC !":GOSUB10
    0:R4=2:GOTO3000

3100 PRINT" WOW! CAN";R$;"RUN. WHAT AN EXHIBITION OF BL
    INDING SPEED. UNFORTUNATELY IT OCCURRED AS A RESUL
    T OF A BLISTERING DISCOVERY CONCERNING DRAGONS AND
     IN THE OPPOSITE DIRECTION OF THAT OF THE PRINCESS
    .":GOTO4500

4000 PRINTR$;"HAS PULLED IT OFF - THE PRINCESS HAS BEEN
         RESCUED";:IF G>RND(30) PRINT;"  - IS IMMEDIATE
    LY ACCEPTED INTO THE KING'S COURT AND IS ALLOWED T
    O DO ALL THOSE NICE LITTLE THINGS THAT ONE DOES HA
    PPILYEVER AFTER":GOTO4500
```

END of Swords and Sorcery fix.

combination with the letter A gives a code of 1 which is the same as the BREAK key and has the same effect.

David S. Tilton
Manchester, NH

A Word for NEWDOS

It is with pleasure that I comment on three week's experience with Apparat's NEWDOS 80.

We have saved much time by using the DELETE/INSERT and DUPLICATE commands for program lines. These have proven invaluable in making major program modifications.

The improvement to SUPERZAP with the DFS command going right to the start of a file has helped our understanding of machine language and our debugging when setting up new files.

The manual tends to be grossly confusing at times and I sense the production of a new jargon which I don't care for, but overall the program has saved us time, and has saved us far more in money than it cost, even though we have only had it three weeks.

On this basis, which for a businessman is a primary consideration, NEWDOS 80 is an excellent program.

Peter G. Dunn
Sturdivant and Dunn, Inc.
Conway, NH

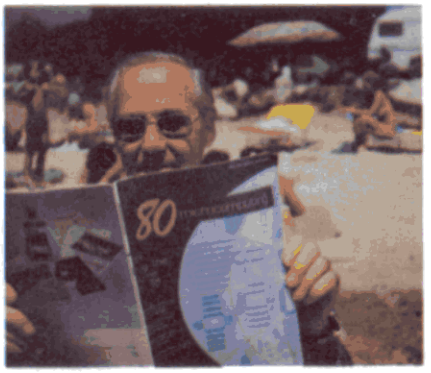
On the Beach

Congratulations on your *80 Microcomputing* magazine and on the excellent quality of the articles and programs, which are improving every month.

To confirm my interest, I enclose a photo at a Saint Tropez French Riviera beach, reading *80* even during my vacation!

As one of your numerous readers, I can assure you of my subscriber fidelity.

J. R. Israel
Paris, France



80 on the French Riviera.

LOOKING FOR MODEL I AND MODEL II BUSINESS SOFTWARE? WE HAVE HUNDREDS OF QUALITY BUSINESS PROGRAMS IN STOCK! AT PRICES YOU CAN AFFORD.

‹WHERE YOUR TRS-80* MEANS BUSINESS›

For the first time you can fill most of your software needs with one telephone call. Whether you are trying to find a specific program, custom software or just help with your system—give us a call.

Invoicing ● Inventory Control ● Accounts Payable ● Accounts Receivable ● Payroll ● General Ledger ● Letter Writer ● Word Processing ● Mailing ● Manufacturing Inventory ● Cost Accounting ● SalesReporting ● Stock Market ● Business Statistics ● Statistical Analysis ● Data Base Systems ● Medical Billing ● Dental Billing ● Special Industries ● Advanced Accounting ● Income Tax ● Language ● Personal Finance ● Technical Programs ● Insurance ● CPA ● Law Office ● Asset Depreciation ● Job Cost ● Utility Programs ● Education ● Games ● Home Programs Loans ● Credit Bureau ● Electronics ● Test Systems ● Sports ● Art ● DOS Systems ● BASIC lessons ● and much more!

Send for our free catalog or give us a call today. We also do custom programs as well as buy top quality programs.

Summer Special:

Complete business system $299.95

OVER 100 OF THE BEST BUSINESS PROGRAMS FOR THE TRS-80* MODEL I AND MODEL II IN STOCK READY FOR IMMEDIATE DELIVERY.

LET US ANSWER YOUR QUESTIONS TODAY.

We now sell:

Structured Systems Group • Graham Dorian • Magic Wand™ • Digital Research, Inc. • Osborne/McGraw Hill • Compiler Systems • Software Mart Software

Software-Mart

286

24092 Pandora St • El Toro CA 92630

In California Call (714) 768-7818 Call Toll Free 1 (800) 854-7115

master charge THE INTERBANK CARD 24 Hour Service VISA

OUR BEST ADS ARE NOT WRITTEN — THEY'RE RUNNING ON TRS-80'S

All Software Mart Programs are sold on an "as is" basis and with "All Faults" Prices and programs are subject to change without notice.

Magic Wand™ is a Trademark of Small Business Applications, Inc.

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.

80 APPLICATIONS

by Dennis Kitsz

"The principle of the clock is simple: Forty times each second, a pulse is sent from the expansion box to the keyboard unit."

Does anybody really know what time it is? Does anybody really care?" Personal computers were still half a generation away when Chicago sang those words, but the phrase has a special kind of relevance for TRS-80 users. How can you get the time when you want it, yet prevent it from intruding when you don't need it?

Your 80 can tell time in two ways. Firstly, an 80 can be forced to keep track of certain predictable events, and update an internal clock program; and secondly, it can read an external clock which ticks along irrespective of the computer.

Built-in Clock

The clock most familiar to Radio Shack users is that built into the expansion interface, which disk and Level III users can print by means of the TIME$ command. The principle of this clock is simple: Forty times each second, a pulse is sent from the expansion box to the keyboard unit. The pulse enters via the interrupt line, causing the computer temporarily to set aside its activities in order to update the seconds, minutes, hours, days, months and year. When you ask for the TIME$ command, the computer merely looks to the area of memory in which it has stored this updated information, and sends it to the screen or printers.

This method is easy to use, but there are some problems with it. Naturally, you need the expensive expansion interface and special software to use the 25 millisecond interrupt. To keep accurate track of the time and date, your computer must remain on 24 hours a day. Without disk, special tape software must be loaded if an inadvertent reset should occur. And don't forget that a CMD"T" must be executed before every CLOAD and CSAVE.

But more important than any of these, is the deleterious effect the 40-times-per-second interrupt and update can have on a program is execution. To keep track of the real time, the interrupt method steals valuable program time.

This month's column will present two inexpensive alternatives to the expansion box clock. The first of these is based on a once-per-second interrupt, reducing execution-time overhead to a reasonable amount. The other system uses a new integrated circuit clock chip, MSM5832, manufactured by OKI and available for $9.80 from Digi-Key Corp. (P.O. Box 677, Thief River Falls, MN 56701, 800-346-5144).

Before we start wiring, let's find out what's important to know about computer clocks. The primary consideration regarding the clock itself is accuracy, especially if it is to be used by the computer to control external machinery, for example, which is critically time-dependent.

The expansion interface clock uses a quartz crystal time base, which is accurate to .001 percent. This percentage tells us that after 100,000 seconds (about one day), the clock will be fast or slow by one second.

But, there is an even more accurate source: the power line itself, which, because it is linked into a large network of generating systems, must maintain a virtually absolute synchronization over the long term of 60 clock cycles per second. Short-duration lags and leads may appear, but the percentage of error over a year is negligible.

Ancient History

Other things to consider are 60 seconds to the minute, 60 minutes to the hour. That's our legacy from the Babylonians and their base-60 number system, so the computer clock has to remember that 59 plus one carries into the hundreds place. Then the Caesars gave us Julius and Augustus to deal with, meaning a duodecimal year, and some irregularly numbered months also turned up.

And finally, a pope named Gregory is remembered because the calendar bearing his name dramatically dropped a few days right out of the middle of the sixteenth century and left us with an uncomfortable phenomenon known as leap year. Because of the vagaries of personal pride, tradition, astronomy and Renaissance, no number system, not even hexadecimal or octal, can compare in complexity with our very own calendar.

Fig. 1 presents a very simple interrupt-driven clock. You'll find no provision for battery backup, since clock updating is done by the computer—no power, no TRS-80. The transformer is 6.3 volts (Radio Shack #273-1384 will do fine). It both powers the circuit and provides the 60-Hz pulse to the system. The sine-wave pulse is shaped into a neat digital signal by Schmitt Trigger Z1. Z2 divides the signal by 12, providing five pulses per second, and Z3 then divides that by five. We are left with a one-second pulse at the output of Z3.

This signal isn't useful exactly as it is, though. Why? When the computer receives any interrupt, it sets aside its current activities and, via a specified program, services that interrupt. Within a few microseconds it's done with that process, and it tries to return to the main program. But the divided-down, one-second pulse is too long. It's on for one-half second, then off one-half second, which means that it will still be on when the computer returns to the main program. The computer, being ignorant and slavish, will bounce unquestioningly back to the interrupt routine and update the time again. And again. And again—until the pulse turns off. By this time, the clock is probably telling you it's tomorrow.

To remedy that problem, the one-second pulse is fed into a flip-flop. The computer provides a very useful handshaking signal called an "interrupt acknowledge" (INTAK), which in effect says, "Okay, bud, I got yer order. Now lay off." So when the clock's interrupt pulse goes on, INTAK immediately resets the flip-flop, cutting the interrupt off; the CPU updates the clock, and is able to return to the main program. It is not again disturbed until the one-sec-

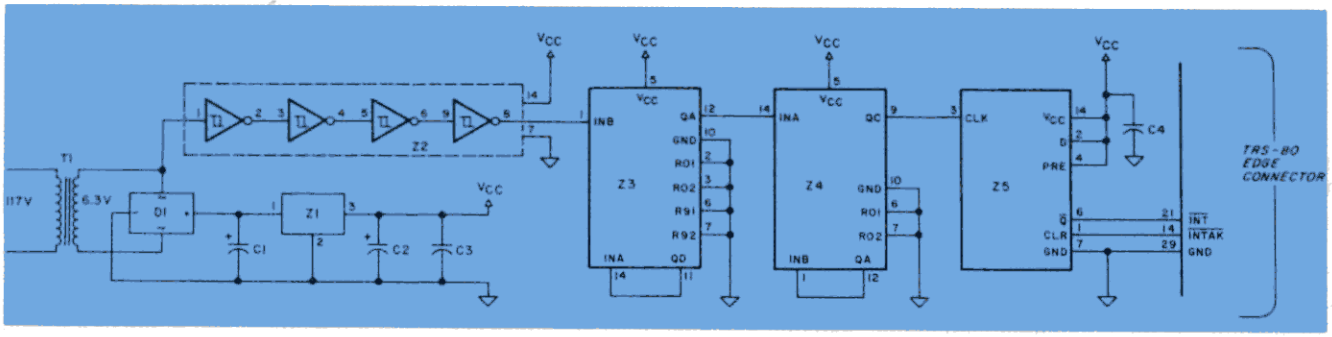


Fig. 1. Real-time Clock Using One-second Interrupts

Enjoying 80 MICRO ? then read on...

80 MICROCOMPUTING has proven, in its first several issues, that it can give you more information on the TRS-80* than any other single source. The magazine has grown more informative with each month and we still have lots more interesting ideas in the works for you.

With the TRS-80* (or 90...etc.) being the most popular microcomputer in the entire world, you are going to benefit from this in many ways. The more computers there are out there of one kind...the more good programs you are going to have for this system. I hope that is obvious. You may be sure that *80 MICROCOMPUTING* will be packed with the shorter programs and reviews of the larger ones. You can waste an awful lot of money on stuff that looks great in the ads, but fizzles out when you try to use it. You need our reviews.

The wealth of programs will also mean that there will be much better programs for the TRS-80* than any other system. Put yourself in the seat of a computer programmer and you'll understand this. If you are going to spend several months developing a comprehensive program, and it takes all of that to write and debug a big program, would you write it for a system which has sold one hundred units or one which has sold over 300,000 systems? The answer is obvious...and this is why we are already seeing programs coming out for the TRS-80* which are far better than anything for any other system on the market. This is tough for other systems ...the law of the computer jungle.

Between our connections with Instant Software, the largest publisher of microcomputer programs in the world, and Kilobaud MICROCOMPUTING, you know that *80 MICROCOMPUTING* is going to be your most important link with software for the TRS-80*.

With Instant Software being sold and promoted in every country in the world where the TRS-80* is being sold, our input of programs is also the best in the world. We get programs submitted from everywhere...often from 50 to 100 a week! You'll get the cream of the crop either published or reviewed in *80*.

HARDWARE TOO

The same law of the computer jungle holds for hardware. Would you, as a manufacturer, market an accessory for a system which has sold 100 units or would you go first for the one which has sold hundreds of thousands. It is, as with software, self-evident why the great bulk of the hardware accessories for computers are for the TRS-80* these days.

80 MICROCOMPUTING has the advantage of the use of the largest and most complete microcomputer lab in the world...the one developed for Instant Software and Kilobaud MICROCOMPUTING. This means that most new pieces of equipment are tested and in use by our staff...and this means that we can tell you what we think is outstanding...and where we find ripoffs. This lab is important to you.

SUBSCRIBE

If you are not already a subscriber to *80 MICROCOMPUTING*, please get signed up right now. The yearly rates are $18, and that is a bargain. Just one single program of use to you can be worth much more than that. One review of an accessory could save you many times that much investment. I would appreciate it if you would appoint yourself a committee of one to get more subscribers for the magazine. You will benefit even more than we do here at the magazine...because the more readers we have, the more ads we will be able to attract...and the more ads, the more pages of articles you will get every month.

The *80* market can, I think, support a couple of hundred pages of ads...and that would mean a magazine of nearly 500 pages a month. That should hold you. You may not have time left to use your computer.

ENCYCLOPEDIA

If you've read Kilobaud MICROCOMPUTING, you know that I try hard not to duplicate published material. My concept is that every reader should save every issue (we sell inexpensive boxes for this so they can sit on your library shelf) and treat the magazine as a continuing encyclopedia of computing. I make sure that much of the material in each issue is written in simple language so it will be understandable by even the rawest newcomer to computers. Oh, I have articles for the more advanced users too, so you'll have something to look back over later and use as your understanding of your system grows.

Try to think of *80 MICROCOMPUTING* as more of a large club newsletter than an ivory tower high-level publication. I'll leave the pomp to other publishers...the ones with the well-deserved inferiority complexes who cater to their inadequacies by publishing esoteric baloney. This magazine is written by the readers and edited by people whose aim is to help you enjoy your TRS-80*.

SAVE

With each issue costing $2.50 at your computer store, that's $30 a year. For $18 a year you can subscribe...at least for now. As the magazine expands, please do not be surprised if the cover price increases, along with the subscription price. I started *73 Magazine* for radio amateurs twenty years ago with a cover price of 37¢ (two for 73¢) and it is up to $2.95 a copy now (and it is the largest of the ham magazines).

For you bargain hunters...and those who find that one year goes by all too rapidly, the three year rate for *80* is $45. This, too, will be going up...reflecting the inflation, paper increases, postage increases, and a short vacation for me in Hong Kong next year. Someone has to pay for that.

*TRS-80 is a trademark of Tandy Corp.

If the coupon below has been used, please fill out subscription form on the Reader Service card in the back of the magazine

YES! Sign me on as a subscriber to 80 Microcomputing for only $18 a year!

Card # _____ Exp. _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

- ☐ 12 issues – $18
- ☐ 36 issues – $45
- ☐ Please bill me
- ☐ Payment Enclosed
- ☐ Master Charge
- ☐ VISA
- ☐ American Express

80 microcomputing™
Peterborough, N.H. 03458

Subscription begins with next published issue.
Back issues, while available are $3 each.
Canada: $20 per year US funds.
All other foreign subscriptions: $28 one year only.

30886

ond pulse again trips the flip-flop.

Round-the-clock Time

You can build this real-time clock for under $10 (cheap enough so you can toss it out if you purchase an interface). But for a few dollars more, the luxury of round-the-clock time is available. The secret is the OKI clock/calender integrated circuit, which is set up for use with microcomputers instead of LED readout digits. To make it work you will need a 32.768-KHz watch crystal (also sold by Digi-Key), a port chip available at Radio Shack, and some simple logic.

The clock chip provides just about all the features we might need: time in hours (12 or 24-hour format), minutes and seconds; month, day and year (even leap year); and day of the week. Beyond the basics, the MSM5832 can provide timing signals to the computer 1,024 time per second, once per second, once per minute, or once per hour. A battery backup of just 2.2 volts will keep it timing when the rest of the system is off. Fig. 2 presents the complete timekeeping circuit for the TRS-80. (Since 74LS260's were in short supply when this article was being prepared, Fig. 3 is an alternate for that part of the circuit.)

The main disadvantage of the integrated circuit clock device is its technology. At present, low power consumption (for battery backup), high density (to squeeze the clock/calendar on the chip), and high speed (to be compatible with fast-moving microcomputers) are not all economically possible.

Although the MSM5832 is billed as a microcomputer-oriented clock, it is not directly compatible with the TRS-80. Intermediate logic must be used to latch onto the clock information in its own good time and feed it to the 80 as the computer's signals speed by.

The INS8255 peripheral interface device does this job, setting up a slower, "private" bus between itself and the MSM5832. The data flows to and from the clock through the 8255's port A, the clock's address (for seconds, minutes, etc.) is selected through port B and special timekeeping features are commanded through port C.

The 8255 circuit is wired so that the clock can be placed at an appropriate place in the TRS-80's memory map. The Z-80 microprocessor has 16-pin connectons which are used to produce an address, or specific memory location. The highest number that 16 lines can produce is 1111 1111 1111 1111 in binary or 65,535 decimal. Within this universe reside the different types of memory used by the 80.

BASIC ROM uses the lowest 12,287 bytes of memory (0000H to 2FFFH). The keyboard needs only eight bytes, but because of electronic and software design convenience, actually takes up an area 1,024 bytes long (3800H to 3BFFH). Cassette, printer and disk eat up a few bytes from 37E0H to 37FFH, and the video screen has a 1,024-byte block of memory reserved for its own use at 3C00H to 3FFFH. A full complement of RAM takes up 49,152 bytes from 4000H to FFFFH.

That's a total of 63,504 bytes. What has happened to the remaining 2,032? They are blank, ready for crafty TRS-80 users to put them to work.

But we have to fit our clock in place carefully, if only because some manufacturers (Exatron with their Stringy Floppy and Personal Computer Products' REX-80) have beat us to it, using this blank block for their own ROM.

Besides, in the near future this column will present a way of adding 2K of your home-programmed ROM to the system from 3000H to 37CFH. So we will place the clock at 37D0H, 37D1H and 37D2H, out of the way of the operating systems mentioned above, and just below the cassette, printer and disk addresses. In Fig. 2, Z1 and Z2 do the decoding work, permitting our access to those 13 addresses only (See Table 1 for details).

Complete software for both clocks will be presented next month.

Assembling the Clock

Wiring the interrupt-based clock is very simple, and, except for the transformer, it can be done on a single board, the RS #276-170 (Fig. 1).

For the interrupt-based clock, only three contacts of an edge connector are used, so you might consider mounting the entire circuit inside the TRS-80 case. The OKI clock/calender is larger and needs many of the computer lines, so an edge connector is imperative.

Cable connectors for the TRS edge card can be purchased from several sources, including Digi-Key and Advanced Computer Products (P.O. Box 17329, Irvine, CA 92713; 800-854-8230), and cost about $12.

Those with more patience and less cash can purchase Texas Instruments 40-pin connectors with 0.1-inch spacing. Digi-Key sells these $2.76 connectors, which can be combined with inexpensive multi-conductor cable sold by BNF Enterprises, formerly B & F, Peabody, MA. Extender cables are available to connect any number of components to the TRS-80 bus from Exatron (3555 Ryder St., Santa Clara, CA 95051; 800-538-8599). A two-for-one cable is $15, and each additional connector attached to a cable is $5.

The MSM5832 circuit is a bit more complicated than the basic interrupt clock and sockets should be used for Z3 and Z4. The clock chip is one of those static-sensitive circuits which should be handled carefully, even though it employs internal protection. Most of all, make sure you do no experimentation or testing with the power connected. Complete the circuit. A

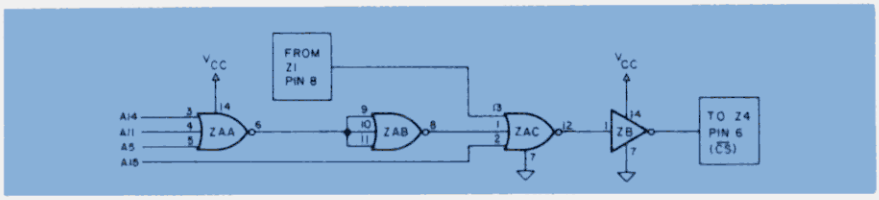


Fig. 2. Real-time Clock Using MSM5832 Clock/Calender

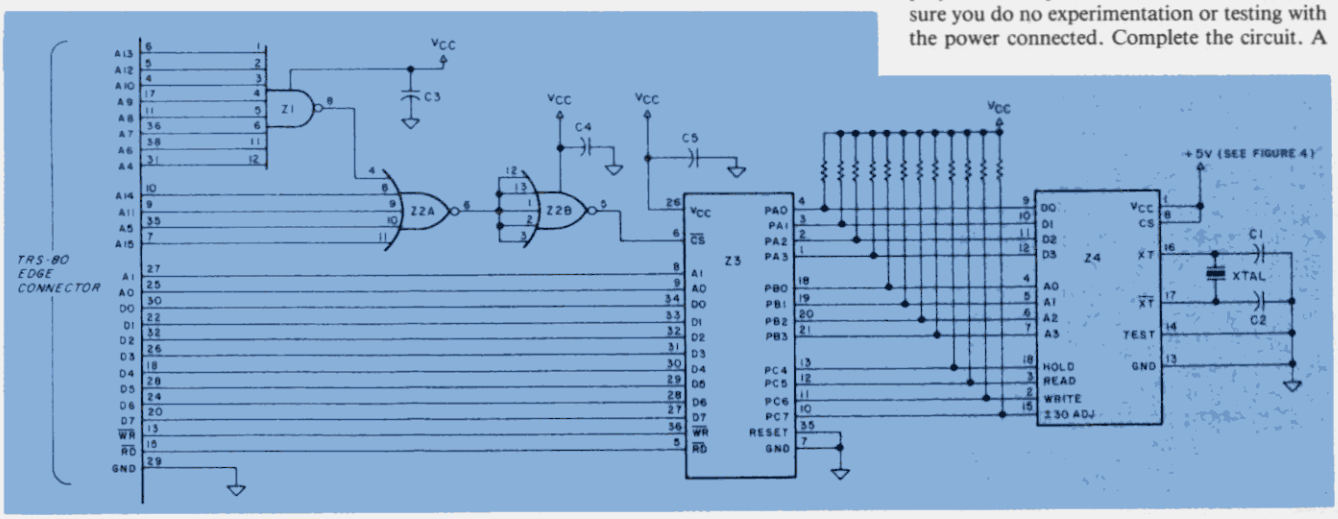


Fig. 3. Real-time Clock Using MSM5832 with Alternate Circuit to Replace 74LS260

How To Decode The Clock Address

Write the address in both hex and binary and determine the corresponding Z-80 address lines:

3				7				D				0			
0	0	1	1	0	1	1	1	1	1	0	1	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Rules:

1a. To create 37D0, eight of these bits (address line) must be 1. Four of the bits must be 0.
b. The lowest four bits will change as we select 8255 ports A, B, or C (37D0, 37D1, and 37D2).
2a. The 8255 integrated circuit is turned on when any address from 37D0 to 37DF appears. This "chip select" (CS) signal must be 0 for the 8255 to respond.
3a. The smallest possible number of IC chips should be used.
b. Only chips that someone manufactures should be used.
c. The chips you need will always be out of stock. Prepare options.

Solution: ●●●●●

1 There is no unique 37D0 address decoder made, so don't look for one.
2 Instead, remember the rules of logic:

INPUT		Type of Gate				
A	B	AND	NAND	OR	NOR	XOR
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

3 Put the "1" lines (13, 12, 10, 9, 8, 7, 6, and 4) into a plentiful and cheap eight-input NAND gate (type 74LS30). Result: a "0" output.
4 Put the "0" lines (15, 14, 11, and 5), plus the above "0" output into a five-input NOR gate (type 74LS260). Result: a "1" output. That decodes all the lines, and it's pretty close to what we need.
5 Put this "1" output into an inverter to change it to the "0" needed to trigger CS. Wait, now, don't get a separate inverter chip. Since the 74LS260 has another five-input nor gate on board, send the previous "1" output into all five inputs of this gate. Voila! Instant inverter.
6 Use Fig. 3 when 74LS260s are out of stock everywhere you call.

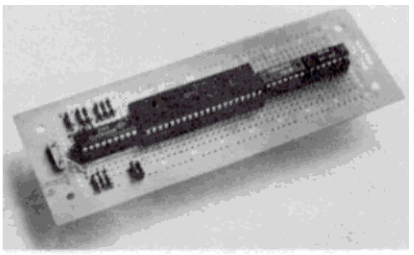


Photo 1. The complete clock/calendar can be built on a small circuit board. Notice the 32.768 KHz crystal on the far left; it is attached to the board with instant glue.

difference of 0.3 volts between certain pins of this chip can be deadly.

This board can be wire-wrapped or soldered, though use great care, little heat and short leads when soldering the crystal in place. Photo 1 shows the entire circuit on a #276-170 circuit board.

Very important to the proper operation of these clock circuits is a ground connecting the computer at edge card pin 29 to the peripheral device's ground, as in the schematics. I have seen several home-brew devices which would function in most frustrating manner until this ground was put in place. Also, take note of the capacitors between Vcc (collector-voltage sup-

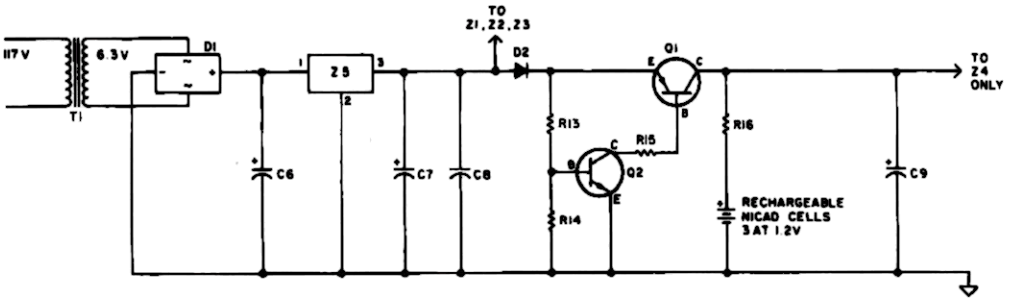


Fig. 4. Power Supply with Battery Backup

The editors plead total insanity in the matter of leaving Dennis Kitsz's photos out of last month's Applications. Space was the first consideration. We'll try to make amends in a future column.—Eds.

FUNCTION TABLE

ADDRESS INPUTS				INTERNAL COUNTER	DATA I/O				DATA LIMITS	NOTES
A_0	A_1	A_2	A_3		D_0	D_1	D_2	D_3		
0	0	0	0	S 1	*	*	*	*	0 ~ 9	S_1 or S_{10} are reset to zero irrespective of input data $D_0 \sim D_3$ when write instruction is executed with address selection
1	0	0	0	S 10	*	*	*		0 ~ 5	
0	1	0	0	MI 1	*	*	*	*	0 ~ 9	
1	1	0	0	MI 10	*	*	*		0 ~ 5	
0	0	1	0	H 1	*	*	*	*	0 ~ 9	
1	0	1	0	H 10	*	*	†	†	0 ~ 1 0 ~ 2	D_2 = "1" for PM, D_3 = "1" for 24 hour format D_2 = "0" for AM, D_3 = "0" for 12 hour format
0	1	1	0	W	*	*	*		0 ~ 6	
1	1	1	0	D 1	*	*	*	*	0 ~ 9	
0	0	0	1	D 10	*	*	†		0 ~ 3	D_2 = "1" for 29 days in month 2 D_2 = "0" for 28 days in month 2 (2)
1	0	0	1	MO 1	*	*	*	*	0 ~ 9	
0	1	0	1	MO 10	*				0 ~ 1	
1	1	0	1	Y 1	*	*	*	*	0 ~ 9	
0	0	1	1	Y 10	*	*	*	*	0 ~ 9	

(1) * data valid as "0" or "1"
blank does not exist (unrecognized during a write and held at "0" during a read)
† data bits used for AM/PM, 12/24 HOUR and leap year
(2) If D_2 previously set to "1", upon completion of month 2 day 29, D_2 will be internally reset to "0"

Address Decoding Table for Z1/Z2

ply) and ground in both circuits. These should be mounted as close as possible to their respective integrated circuits so they can filter out extraneous signals caused by fast-switching ICs. The pull-up resistors (R1 to R12 in Fig. 2) are essential to the operation of the MSM5832. See Fig. 4 for a diagram of the power supply with battery backup for the MSM5832 clock.

Next month's column will cover machine language and BASIC software for using these real-time clocks.

Letters about Applications

Some unexpected difficulties can plague those of us working on hardware additions to the TRS-80. Ron Gillen of Hustisford, WI writes to say that TRS-80 "edge card connectors are not created equal...I purchased a 40-pin edge card connector to replace the failed (cheap) Stringy-Floppy 'Kel-AM' connector. A much more rugged Alpha Mfg. connector was used and, to my dismay, the pins on the edge card side are reversed as pairs. That is, Kel-Am places pin 1 at the top of the edge card and the particular Alpha that I purchased places pin 1 at the bottom of the edge card. This reversal of pairs, which I did not suspect, caused strange and interesting results on the TRS-80 when connected to the ESF cable."

I had the same problem as Ron while building a prototype for this column, so check those cables carefully.

●●●●●

Several readers have asked where to get parts for projects described in this column. In most cases, parts are readily available from a local Radio Shack. Where special parts are needed, the names and addresses of suppliers will be listed. This question was prompted by the interface published earlier this year. By the time the article appeared in print, Radio Shack had discontinued the 81LS95 and 81LS96 circuits. For those still interested, these ICs may be obtained from any of the mail order firms mentioned in this column.

Will readers who have built or plan to build the device please drop me a post card (Roxbury, Vermont 05669)? Future columns will refer to this small interface if readers have built it and find it worthwhile.

●●●●●

Mea culpa, indeed. Chuck Lingo writes from Gardner, KS about those BASIC commands accessible using Seespot! (July):

Your article said all BASIC commands, yet there are only 64 graphic characters and 86 possible key combinations. Where are things such as CHR$, STR$, LEFT$, RIGHT$...etc.?"

An earlier version of the Seespot program being developed made those extra commands available, but I lost them along the pathway to user convenience. Sometime soon this column will present a quick way of evoking all those BASIC commands via single keystrokes.

●●●●●

A quick note on the vagaries of electronic developments: This month's topic was well on its way to completion using some entirely different hardware. Until March, when OKI officially introduced the MSM5832, there were only standard clock/calender chips on the market.

In order to make them microcomputer compatible, the seven-segment LED display outputs had to be converted to binary, using either a home-programmed PROM or a special-purpose National Semiconductor IC. The clock chip's digit strobe was converted into an addressing circuit, and some fancy electronic spaghetti allowed this entire mass to act as a computer clock.

It also cost a lot of time, more money—$30 and having a friend who could program small PROMs. The OKI data sheet walked through the door at just the right moment, promising to be the perfect solution. But that solution is still some time away, when a computer-bus-compatible, high-speed clock chip finally appears on the market. ■

EDUCATION 80

by Earl R. Savage

The first question of the new 80 owner is: "How and where do I get programs?" You may be a teacher carrying your personal 80 to the classroom. You may be a parent wishing to supplement your child's schoolwork. Perhaps you are in charge of a learning lab with several 80s, or a parent wanting to get your preschooler off to a good start. In any case, there is always a search for good programs.

That need arises again and again when talking with educator-users. Unfortunately, there is no simple answer because the sources are many and, at the same time, few.

At this point in time, there are several ways to acquire instructional programs:

● Write your own programs; this is probably your very best source. Better than anyone, you know what you want to teach and you know the student(s)—where he needs to begin, how he learns best and so on. If you don't know the subject well enough—small engine repair, for example—collaborate with a colleague who does.

This program source is the least expensive but it does take time, a commodity that most of us have in short supply. Since your time is not unlimited, why waste it reinventing the wheel? Write what is *not* available and get what someone else has already invented.

● Copy programs from magazines and books; program listings can be found in many books and magazines. It is true that most are not instructional, but if you look carefully, you'll be surprised how many fall into that category. In addition, you will find others that you can modify in one way or another to turn them into teaching programs.

● Exchange programs with others; this can be a really fine source of programs. You will get a bummer now and then, but that happens

1. Outlasts every competitor—200,000,000 character head warranty
2. No duty cycle limitations—even in demanding business applications
3. Professional print quality—9 x 7 matrix
4. Rugged business use construction—metal chassis—two motors
5. 80 characters per second
6. Upper and lower case—full 96 character ASCII set
7. Double width characters
8. Connects directly to TRS-80,™ APPLE® and other computers
9. Block graphics—64 shapes for charts, graphs, diagrams
10. Friction and pin feed
11. Plain paper—up to 3 parts
12. 6 and 8 lines per inch—program controlled paper savings
13. 80 and 132 columns—program controlled
14. Price—the best value in the industry. Call or write today for the name of your local Microline 80 dealer.

14 REASONS WHY TRS-80™ OWNERS CHOOSE THE MICROLINE 80

All fourteen are standard with every Microline 80. The only options are snap-on tractors and a buffered (up to 2000 characters) RS232 interface.

OKIDATA ↗245

Okidata Corporation
111 Gaither Drive, Mount Laurel, New Jersey 08054
Telephone: 609-235-2600

↗ Reader Service—see page 226

when you buy them, too. Exchanging what you have written for what someone else has written is a good and inexpensive way to build up your program library. The cost is limited to your blank tape, postage and, in some cases, a nominal fee.

There are some educationally oriented exchange groups on my present list. I'll share the names and addresses of any exchange groups that send me information about themselves *provided* that they refuse to exchange copyrighted programs. (I won't help them break the law.) If you represent such a group, send me your list and your rules.

● Buy the programs you need; there are some competent and reliable people out there who are writing and selling educational programs. I am also painfully aware that there are some incompetents and/or crooks doing the same thing!

There you have the methods of acquiring educational programs: write them, type them, exchange them and buy them. Do some of each. And don't forget that you can often modify a program to improve it or make it useful for another purpose.

Educational Review: Three Program Books

As I mentioned above, one of the less expensive ways to get programs is to type them from listings in books and magazines. It takes some time and care but the cost is much less than buying them on tape. Here are three books in which you may be interested.

● *57 Practical Programs and Games In BASIC* (Ken Tracton; Radio Shack #62-2008)

The presentation of these programs includes a brief description, any math formulas used, a sample run, a list and a flowchart. As the title indicates, the programs are in BASIC. Most will run on Level II as written, but a few require modification. For Level I users, more extensive modifications are needed.

Though this volume is well written, there is little here applicable to education except for the advanced math student.

Contents: mathematical functions (42); games (8); electronics (3); miscellaneous (4).

● *80 Programs For The TRS-80* (Perry and Brown, editors; 1001001 Inc., Peterborough, NH 03458)

This book was offered originally as a bonus to charter subscribers of this magazine. The listings are in small print but there are plenty of them. All but a few of the programs will fit into a 16K machine. You will find a variety of educational programs here.

Contents: instruction (15); business (13); games (15); utility (6); personal use (10); energy conservation (3); amateur radio (6); electronics (4); miscellaneous (8).

Instruction breakdown: electronics (1); math (6); social studies (4); reading/spelling (3); music (1).

● *TRS-80 Programs* (Rugg and Feldman; Radio Shack #62-2064)

This book contains 32 programs for Level II. All will fit into a 16K machine and most will fit into 4K. Each program is explained in thorough detail including suggested modifications for different uses. The program discussions are so complete that this volume is excellent for study by example, to improve your own program writing. One of the instructional programs (Flashcard) can be used with many subjects and grades.

Contents: instruction (7); games (13); personal use (2); business (1); math functions (9).

When you are typing a listing into your 80 be *very* careful—go slowly and check often what you have entered. You know that the smallest mistake can prevent a program's running. Especially watch out for the letters I and O as compared with the digits 1 and 0.

If you have several programs to type, try to talk a friend into typing half of them. You can then exchange cassette copies and each of you will have saved half the time. In any case, it is a good idea to take a typing break now and again; being tired leads to *misteaks*!

Recording Scores

You may want to record your student's score on an instructional program. If you have even the simplest printer, you don't have to copy scores manually from the display. The following listing will do the job on the Radio Shack Quick Printer II. Change the commands to suit your printer:

```
1030  PRINT "PLEASE SWITCH ON THE
      PRINTER AND THEN PRESS ENTER."
1035  IF INKEY$ = ""THEN 1035
1040  LPRINT "NAME : "A$  :'NAME
      PREVIOUSLY ENTERED
1045  LPRINT "PROGRAM : AMATEUR
      THEORY II"
1050  LPRINT "SCORE : "R"RIGHT OF "T"
      ATTEMPTED"
1055  FOR X = 1 TO 4 : LPRINT : NEXT
1060  END
```

The variable names will have to be changed to agree with those in the program. If you're wondering about line 1055, that simply runs the paper up four lines so that the printout clears the tearbar.

This small section prints the essential information and is easily expanded to include whatever you want on the record. Best of all, not only can you put it in programs that you write, but you can insert it easily into any you have written or any that you have bought.

A Reminder

Don't forget to send me information on your program exchange group so I can pass it along to other readers. And let me know of any special topics you would like discussed in the future. ■

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

Last night, a warm Sunday evening, while my dog and I took our customary walk, I bumped into my next door neighbor. He looked harassed and down in the dumps. On inquiry, he told me a sad tale of his slaving over the books in his office all weekend long. His major problem was getting the payroll recordkeeping up to date in order to file his quarterly returns.

His was a sad, but familiar tale.

CP/M and CBASIC

Radio Shack has recently released the Model II Payroll package (Catalog # 26-4503). Although this effort has some flaws (such as no New York city withholding computation), it represents a substantial piece of work. It can certainly be considered for locations where a city tax is not required or can be calculated as a percentage of gross pay or federal tax.

In addition to the Radio Shack package I will also evaluate the Structured Systems Group, Inc., (SSG) CP/M (Control Program for Microcomputers) payroll package.

It's no secret that TRSDOS has many disk operating system competitors. The most popular of the alternate systems is CP/M, developed by Digital Research, Pacific Grove, CA. CP/M was established as an industry standard long before the Model I TRSDOS was a working system. In fact CP/M was almost adopted as the Model I operating system and was licensed by Tandy although it was never released.

The major problem with CP/M in the Model I was that standard CP/M required low memory in order to work. This area was already used by the Level II ROM. CP/M had to be rewritten to function above the ROM.

Because CP/M has been readily available, many firms have developed software to integrate with it. One such firm, Software Systems, Inc., developed a BASIC interpreter called CBASIC. CBASIC differs from standard Radio Shack BASIC by requiring a compilation phase before execution. At the conclusion of the compilation phase, a special program is created that can be executed but not listed. It is interpreted by the run-time module of the CBASIC system. In this way vendors of CBASIC programs can retain control of the source code. This means that purchasers of CBASIC programs cannot modify the programs without the assistance of the vendor.

SSG Payroll

One of the vendors using CBASIC and CP/M is SSG of Oakland, CA. Their payroll system is distributed by computer stores throughout the country and is part of a total integrated accounting system. Although the system is distributed on three single-density eight-inch disks, it will run on a two-drive Model II. Drive A (equivalent to drive 0 in a TRSDOS system) contains the system and drive B (equivalent to drive 1) contains the data. The

Transform your programming language into a real data base management system

*SPECIAL NOTE! HDBS interfaces with CBASIC under CP/M, with TRS-80 (Models I and II), and with APPLE DOS.

A major breakthrough in data handling for micros! At best, existing programming languages provide only rudimentary file handling capabilities. With HDBS (Hierarchical Data Base System), you can now *transform your programming language* from an ordinary file handling system into a flexible, powerful, and productive data base management system.

HDBS provides the two components any genuine data base system must have:

1. The Data Description Language (DDL), an easy-to-use, stand-alone language for specifying blueprints or schemas of data base organization.
2. The Data Manipulation Language (DML) for manipulating (storing, accessing, modifying, deleting) data organized according to a data base blueprint. The DML extends your data handling capabilities by allowing you to embed powerful data manipulation commands in the programs you write. The commands are stated through the "CALL" facilities of your programming language.

HDBS users are . . .

- free of the need to design files.
- free of the need to merge files; there is no scattering of data over several files. Records of many different types are automatically maintained in a single, integrated organization . . . one that can be spread over numerous disk drives.
- free of the need to be concerned with disk I/O because it is handled automatically by HDBS.

HDBS offers . . .

- hierarchical schema designs.
- data base schemas of up to 254 record types; each record type may contain up to 255 fields. The size of a field may be up to several thousand bytes in length.
- data bases spread over one to eight disk drives. HDBS is independent of the sizes and types of drives.
- user-defined names for fields, record types, and sets.
- records maintainable in several sorted orders . . . and in other orders as well.
- *written in machine language for maximum execution efficiency and minimal memory usage.*
- available versions: Z80 (requires approx. 18K), 6502 (approx. 26K), 8080 (approx. 22K). Total memory requirement must allow for buffer areas.

For more information on the HDBS, write or call us today!

NEW!

HDBS can be used to extend any of the following programming languages under the indicated operating systems:

CP/M with CBASIC; Microsoft BASICs, FORTRAN or COBOL: InterSystem PASCAL/Z; Sorcim PASCAL/M; Micro Focus CIS COBOL; Digital Research PL/I
MVT/FAMOS with BASIC
OASIS with BASIC
TRSDOS and NEWDOS (Models I and II) with *Disk BASIC*
North Star DOS with North Star BASIC
Apple DOS and Applesoft BASIC
Machine Language interface available on all above systems.
Note: Because HDBS can be integrated with a wide range of languages and operating systems, it provides *uniform methods of data handling* across those many languages and systems.

HDBS/QRS. An interactive Report-Writer/Query-System with these features:

- may be customized for non-technical users.
- complex retrieval conditions may be specified.
- detailed reports can be quickly generated.
- wildcard and "match-one" string specifications included.

HDBS/SRS. This Schema Redesign System permits . . .

- renaming fields, record types, sets.
- adding new fields to existing record types.
- allocating additional pages to an existing data base.

Ordering information (applicable to Z80, 8080, and 6502 versions):

HDBS	$300.00
HDBS/QRS	300.00
HDBS/SRS	150.00
HDBS/QRS/SRS Package	675.00
HDBS expansion to MDBS*	650.00
HDBS Manual	35.00
QRS Manual	5.00
SRS Manual	5.00
System Specific Manuals (each)	5.00
Guide to Data Base Management	10.00

*HDBS may be expanded at any time to full network data base management system: MDBS.DMS/MDBS.DDL

Within a given operating system, add $125.00 for each additional language selected.

For prices outside the U.S. and Canada, please ask for price lists.

Add $2.50 handling fee for non-cash order ($5.00 outside U.S.).

When ordering, specify intended use with . . .

1. North Star DOS and BASIC
2. CP/M - CBASIC
3. CP/M - Microsoft BASIC 4.XX
4. CP/M - Microsoft BASIC 5.XX
5. CP/M - Microsoft BASIC or FORTRAN Compiler
6. CP/M - Microsoft COBOL-80
7. CP/M - InterSystem PASCAL/Z
8. CP/M - Sorcim PASCAL/M
9. CP/M - Digital Research PL/I
10. CP/M - Micro Focus CIS COBOL
11. TRSDOS/NEWDOS and TRS Disk BASIC (Models I and II)
12. Apple DOS and Applesoft BASIC
13. MVT/FAMOS and BASIC
14. OASIS and OASIS BASIC
15. Machine Language Programs (Specify operating system.)

Indiana residents add 4%.

We accept Visa and Master Charge.

Micro Data Base Systems, inc. ✓446

Box 248, Lafayette, Indiana 47902
317-742-7388 or 317-448-1616

Micro Data Base Systems, Inc.
Setting standards of excellence for data base software . . . worldwide.

CP/M and CBASIC software required to run the system are not included.

Simple payroll systems usually involve a combination of automated and manual techniques. Automation is generally applied to the mechanical calculation of the withholding taxes and net pay. Check writing and maintenance of employee earnings data must be accomplished by hand. Because many devices used for automation purposes have the capacity to store the payroll data, cumulative totals can be carried forward to permit the preparation of required tax returns.

Payroll preparation does not require complicated calculations. Most of the parameters are defined by federal, state and local withholding charts. Since these charts are usually quite short, they can easily be accommodated in memory and referenced by a table look-up routine. This routine usually converts gross pay to an annualized figure by multiplying it by the payroll frequency. The resulting figure less exemptions is referenced to the various tables to calculate the tax due. The tax is then divided by the same payroll frequency to determine the tax required when calculating net pay. For the current pay period these calculations must be performed for every eligible person on the payroll.

A series of parameters is supplied for each employee to determine which tax table is used and which exemptions are to be applied. Typically these are specified on the employee earnings master file. This master file is usually quite long and complex. For example, the employee master file in the SSG payroll system is almost 700 bytes long.

The need for such large files limits the amount of data that can be accumulated for each employee. For this reason neither Radio Shack's nor the SSG's payroll system have a detailed employee earnings history. Both systems carry cumulative quarter-to-date and year-to-date information only. This can create a problem if it is necessary to know payroll data by employee for specific pay periods. Otherwise, both payroll systems will do an excellent job of keeping payroll records.

Similar Features

The features of the two systems are similar in many respects. Both calculate deductions from employee wages for federal income tax withholding, Social Security, state disability and state and local income tax.

Both systems allow you to preload the employee payroll earnings record with information that remains constant for each pay period. If there are no changes, payroll calculations will be automatic. Default tax rates for FICA and federal withholding tax tables are also provided by both.

The Radio Shack payroll will not handle local income taxes, if withholding is based upon a table. It will only handle local income taxes if the withholding amount is calculated as a percentage of gross pay of Federal Tax.

Radio Shack's Model II Payroll package uses two special function keys, F1 and F2, to control editing and data entry. Because the SSG's package was not designed for a specific function, keys are not enabled. The control key and lettered key combinations are used to allow you to call files or update records.

While the SSG system provides payroll withholding tax tables for New York state and New York City, the Model II Payroll package provides the guidelines to generate the necessary tables for all 50 states.

Payroll data is entered into the SSG system in batches. Although the screen presentation and data entry phase are relatively cumbersome, the number of different items that the system will accept is outstanding. Data such as withholding tax overrides, special payments and expense reimbursements are easily accommodated.

The Model II Payroll's data entry screen displays the full gross to net calculation. Changes are easily made to the necessary fields and the calculation is updated instantly. In addition, there is a time card calculator mode which allows you to input daily hours and calculate the weekly hours right on the screen, which is a great aid to data entry. At the conclusion to the data entry phase, a payroll journal can be printed.

In the SSG system the data that is entered must be sorted and a batch listing prepared before the payroll can be calculated. From a security point of view, separating the data entry function and calculation procedure is better. In the Radio Shack system it is quite possible to edit a payroll after checks are printed, and reprint a single check. It is not possible to do this in the SSG system.

Both systems use a preformatted check. Checks will have to be ordered that are compatible with each system's printing program. Both systems print payroll journals, check registers, and journal entry data. Both can also be integrated into a companion general ledger system.

The SSG program differs in handling payroll expense information. In the SSG system it is possible to arrange employee payroll expense in five categories on a percentage basis. For example, if a employee works half his time in design, and half his time in sales, it is possible to distribute half his earnings from a sales payroll category and the balance from a design payroll category. The Radio Shack Model II Payroll cannot. Regular time, overtime, double-time, vacation and holiday payroll categories are used instead.

Application Limits

The lack of a payroll distribution by cost center in the Radio Shack payroll could limit its application in firms with many payroll cost centers. The only alternative would be to prepare a separate payroll for each cost center, so that payroll expense can be charged to the proper accounts.

However, in environments where an analysis of gross pay is necessary and overtime pay, vacation, holiday and sick pay must be separately identified, the Radio Shack payroll system has a clear edge. In the SSG system these items are not separately identified. They are combined with the FICA and UIT tax expense in determining the total salary charge.

From an accounting standpoint, combining salary and non-salary amounts into a single account is not desirable. To perform a payroll audit the pure salary expense figure must be available for review.

Both systems prepare a full spectrum of quarterly and annual reports. In addition, master file printouts and individual employee records are available showing cumulative earnings paid, along with other year to date statistics. The SSG master file also has provisions for accumulating data, such as accrued vacation and sick pay. These statistics are not available in the Radio Shack payroll. The SSG payroll costs $1,250 and is available from distributors of CP/M systems throughout the country. In order to run the payroll, purchase of the CP/M operating systems and CBASIC is required. Because CP/M for the Model II costs $195 and CBASIC costs an additional $100, the total cost of the SSG payroll system is a shade over $1,500. The Radio Shack payroll system costs less than $400 with slightly fewer features.

There has to be a powerful incentive to spend the additional $1,100 for the SSG payroll system.

The SSG system represents a mature software system which has already been used many times. The Radio Shack system, on the other hand, is a brand new package. Experience with new software products indicates that there will probably be a "shake out" period before all the bugs are worked out: Bugs can contribute significantly more than $1,100 of aggravation in an industrial environment.

An example of the care in the SSG system design is the use of data files with numbered extensions. If a foreign file gets into the system, the internal number checking of the operating system will indicate that an error has occurred. These advanced techniques are generally transparent to the user, but should an error occur, an error message will appear so that recovery procedures can begin.

More Information

Considerably more information is provided about the SSG system than the Radio Shack system despite the fact that Radio Shack's has a source code listing. The SSG system is also designed to accommodate additional custom programs. If you wish to write a program to accumulate employee earning statistics, it can be incorporated and selected from the menu. The structure of all files and a complete input and output tracking of each run is provided for programming. This can also be used to aid restart and error correction procedures.

While there is no doubt that Radio Shack software will also include this data at sometime in the future, the payroll documentation package is not there yet. It is a good bet that as Radio Shack software is distributed to more commercial environments, the documentation and system controls will eventually evolve to the same level as the SSG system.

Although the SSG system represents a more mature, flexible and comprehensive system, the Radio Shack payroll system, if used with knowledge of its limitations, will prove a useful accounting tool as well. ■

NOW YOU HAVE A CHOICE OF SOFTWARE FOR YOUR MODEL I or MODEL II.

Structured Systems Group
Graham-Dorian
Magic Wand™
Digital Research, Inc.
Osborne/McGraw Hill
Compiler Systems
Software Mart Software

Software-Mart

286

24092 Pandora St • El Toro CA 92630

In California Call (714) 768-7818 Call Toll Free 1 (800) 854-7115

master charge 24 Hour Service VISA

OUR BEST ADS ARE NOT WRITTEN — THEY'RE RUNNING ON TRS-80's

All Software Mart Programs are sold on an "as is" basis and with "All Faults" Prices and programs are subject to change without notice.

Magic Wand™ is a Trademark of Small Business Applications, Inc.

If You've Ever Dreamed of Flying

These 7 flight simulation programs can take you aloft, into the realm of powered flight. We're Instant Software—Fly Us!

Night Flight

May, 1941—The dreaded Axis battleship, the Bismarck, has broken out of the North Sea and is now somewhere in the North Atlantic. Your mission is to make a nighttime photo reconnaissance flight over the Bismarck. These photos will help the Admiralty determine the extent of damage done to the Bismarck in a previous battle and whether the British fleet has a chance to sink the German pocket battleship.

The Night Flight program lets you take-off, fly, and land a propeller driven aircraft. You can practice approaches and landings with a full on-screen display of the landing field information. The instructions with this program can practically teach you to fly.

Somewhere out on the cold, gray North Atlantic, the Bismarck tries to elude her pursuers. Your photos are vital. Launch yourself into the night sky with the Night Flight package.

Order No. 0117R $9.95

Air Flight Simulation

Your aircraft is on the runway loaded with fuel, instruments feeding the computer a constant stream of information.

A glance at your flight screen gives you airspeed, altitude, and compass heading. After you take-off, the all important Ascent/Descent-Turn/Bank Indicator will tell you the attitude of your aircraft at a glance, whether you are climbing, diving or banking into a left or right turn.

Your mission is a short one. You have a maximum possible range of about 50 miles, on one precious tank of fuel. Your objective is to take-off, fly the aircraft, and land without crashing.

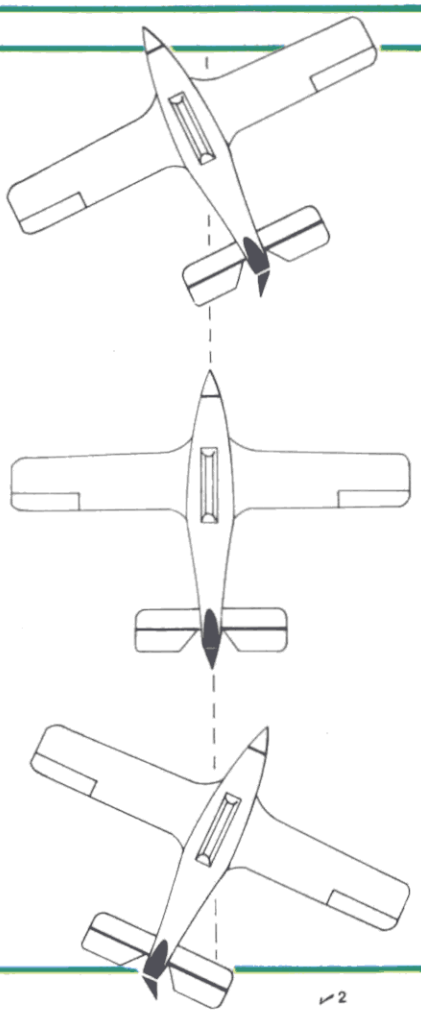
You may not have been at the controls of an aircraft before. The basic flight instructions enclosed will be invaluable. Included are explanations on basic aerodynamics and principles of flight, plus illustrations telling you how to recover from dangerous maneuvers.

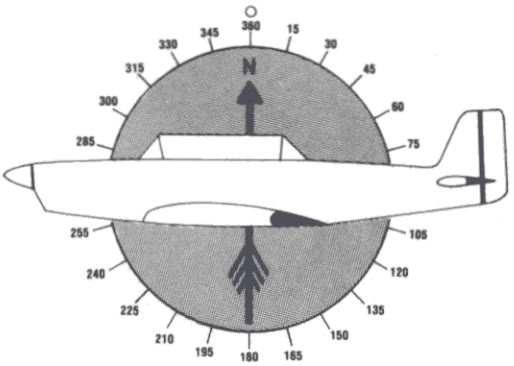
Your aircraft will respond rapidly to the controls, and your movements must be delicate. Too much airspeed and your aircraft could explode from overstress. If the airspeed is too slow, you might stall and crash. A clumsy turn, and you might find yourself flying upside down, fighting to regain control.

It will take a few hours of flight time, before you can take-off and fly a correct flight plan. By then you will be expert enough to attempt aerobatic maneuvers.

With Air Flight Simulation and enough flight time, the sky's the limit!

Order No. 0017R $9.95





Flight Path

The Flight Path package will let you experience all aspects of modern day aviation.

Mountain Pilot transforms you into a daring bush pilot as you fly badly needed supplies to a remote gold mining camp. You'll have to cross a hazardous mountain range, while struggling with headwinds, tricky navigation and rapidly diminishing fuel.

Watch your airspeed, altitude and rate-of-climb or you could stall-out and crash. If you deliver your supplies, you can't relax; you must return over those mountains with a heavy cargo of gold bullion.

O'Hare is a control tower simulation where you become an Air Traffic Controller. The lives of hundreds of people become your responsibility as you guide aircraft through your control sector to a safe landing.

You'll have to deal with different aircraft requirements, wind change warnings and potential midair collisions. But no matter what happens, you must bring in each of the twenty aircraft in your tour of duty.

Precision Approach Radar combines the skills of pilot and Air Traffic Controller. You become the pilot's "eyes" as they try to land in limited visibility conditions. Your commands guide the aircraft in its approach to the field and a safe landing.

The Flight Path package covers both sides of flight procedure, from the thrill of flying to the tense drama of air traffic control.

Order No. 0171R $9.95.

TO ORDER: Look for these programs at the dealer nearest you (see list of dealers on page 206). If your store doesn't stock Instant Software send your order with payment to: Instant Software, Order Dept., Peterborough, N.H. 03458 (Add $1.00 for handling) or call toll-free 1-800-258-5473 (VISA, MC and AE accepted).

Prices subject to change without notice.

Instant Software™

PETERBOROUGH, N.H. 03458
603-924-7296

Jet Fighter Pilot

The Jet Fighter Pilot package takes you as close to real combat flying as possible . . . without pulling G's.

In this brilliantly realistic simulation, you become the pilot of a high performance, twin turbo-jet fighter. Total control of the aircraft is yours.

At the start of your mission, you'll go through an entire engine start procedure before your flight (provided your ground maintenance is up to par). Your takeoff will be from either the deck of an aircraft carrier (via a steam catapult) or from an airfield.

All controls respond the same as they would on a real jet fighter. You'll have to constantly monitor your display and make adjustments to your throttle, flaps, rudder and air spoilers. You decide when to retract flaps, landing gear and release the auxiliary fuel drop-tanks.

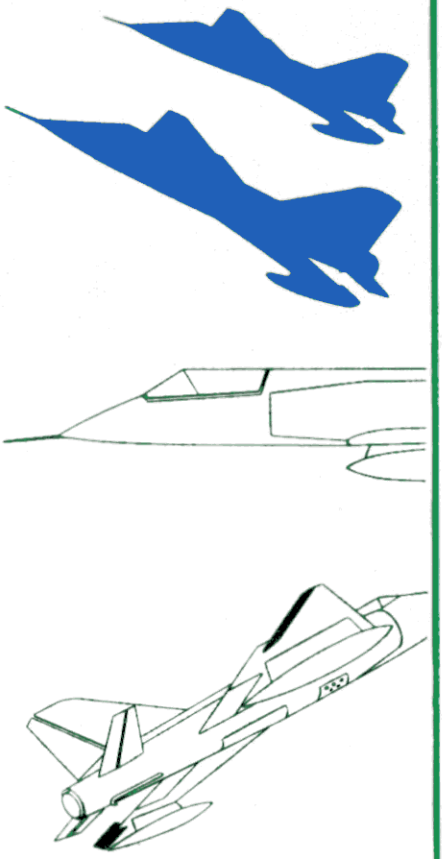
Your on-board navigational computer will direct you to your selected airport. The Glideslope/Localizer information will aid you in approaching and landing on an aircraft carrier deck or airfield.

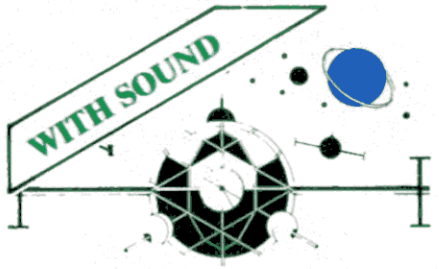
The Weapons Control Computer will arm your missiles, provide you with the range and bearing to a target, and tell you when to attack. And, if things should get a little too hot, you have an ejection seat command for egress.

For a carrier-based landing, you'll have to deploy your tail hook. For a land-based landing, you'll need reverse thrust and your drag chute.

After you've flown a few missions with the Jet Fighter Pilot package, you'll *know* you've earned your wings.

Order No. 0159R $14.95





Ball Turret Gunner

"No personal consideration should stand in the way of performing a public duty."

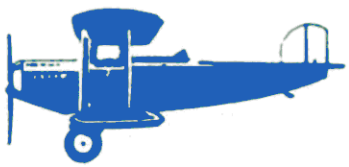
A 19th Century Terran military commander

For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC Exxonerator Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

Imagine yourself at the control console of an LW-1417 Stratoblazer (Type B Strategic Laser Weapon). Your Hindsight Director informs you that a Gnat fighter is coming in for an attack. You pivot your gigawatt laser turret until you can see the target on your monitor. The Range Indicator shows him coming in fast. The Targeting Computer studies his course and speed as your finger tenses over the firing key. You know you'll have only a fraction of a second in which to react. The Gnat fighter's evasive maneuvers cause him to dance in your sights. Suddenly, you see the FIRE Command and you react instinctively. Your laser beam lashes out and reduces the Gnat to an expanding ball of ionized gas. Mission accomplished!

Ball Turret Gunner, with your choice of multiple levels of difficulty, optional sound effects and superb graphics, is more than just a game. It's an adventure. Experience it!

Order No. 0051R $9.95



Airmail Pilot

Let the Airmail Pilot package take you back to the early days of aviation history. Your plane is the Curtis JN4-D, affectionately known as the Jenny. You must fly the mail from Columbus to Chicago.

The Jenny carries only 26 gallons of fuel. You'll have to stop along the way. Bad weather may force you down. Electrical storms may turn your aircraft into a mass of flaming wreckage, or ice may form on your wings and plunge you to certain death below. But, the mail must get through.

The onboard clock will show your elapsed time. Your mission is to complete the flight as quickly as possible.

Experience the thrills of flying, when aircraft were mere fragile machines of wood and fabric, with the Airmail Pilot package. (Scarf and flying helmet optional.)

Order No. 0106R $9.95

Cosmic Patrol

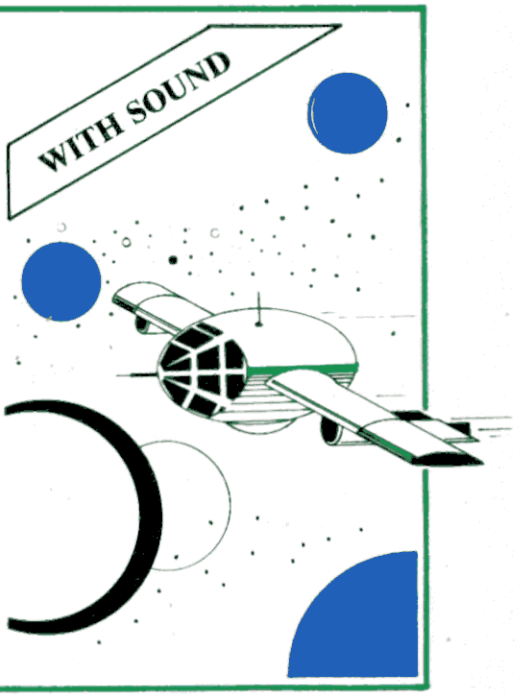
WARNING: PLAYERS OF THIS GAME SHOULD BE PREPARED FOR A STATE OF REALISM HITHERTO UNAVAILABLE ON THE TRS-80

The Cosmic Patrol program puts you in the command chair of a small interstellar patrol craft. Your mission is to defeat Terran space and prey on the Quelon supply ships which carry essential parts and lubrificants for that implacably hostile robotic force. The drone freighters are fairly easy pickings for the accomplished starship pilot, but beware of the I-Fighter escorts. They're armed, fast and piloted by intelligent robots linked to battle computers. They *never* miss.

The Cosmic Patrol program is not just another search and destroy game. With its fast, real-time action, impressive sound option and superb graphics, this machine-language program is the best of its genre.

Don't keep putting quarter after quarter into arcade games or spending big bucks for video game cartridges. Get Cosmic Patrol from Instant Software—and get the best for less!

Order No. 0223R $14.95



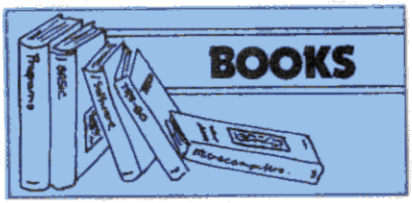
Prices subject to change without notice.

Instant Software™

PETERBOROUGH, N.H. 03458
603-924-7296

80 REVIEWS

"Typical program samples include preliminary comments, fully annotated source code, line by line explanations and descriptive notes."



Inside Level II
John Blattner and Bryan Mumford
Mumford Micro Systems
Summerland, CA
Softcover, 65 pp.
$15.95

by Dennis Bathory Kitsz

Computers like the TRS-80 have somewhat dulled our sensitivities. We accept crude screen imagery and dream of high resolution graphics, which are still a far cry from fine-grained photography.

Four hundred years of design have given us a wealth of elegant typefaces that express the utility of writing and please the eye, yet we are content, even enthusiastic, with the unpleasant dot matrix characters on video and printer. We perceive squawks and bleeps as something reminiscent of music, in spite of composers and performers who cringe at these frigid sounds.

It was a refreshing experience to receive *Inside Level II*, a volume of valuable information about the TRS-80's ROM, yet immensely thoughtful, literate and cleanly designed. Mumford Micro Systems may be a small company, but it isn't a member of the "type it, copy it, bind it in the attic" crowd, represented by many shoddily prepared volumes.

Pioneers though we all may be in this field, we should not permit our aesthetic senses to be bludgeoned. For their artistic efforts alone, Mumford Micro is to be thanked.

Inside Level II is the result of a cooperative effort by John Blattner and Bryan Mumford. It presents 18 chapters of detailed descriptions of and applications for ROM subroutines and entry points.

Blattner and Mumford tell us in their introduction that there is an efficient scheme for linking BASIC and assembly language programs: "The result can be a single, smoothly joined program that combines the best features of both languages - the ease of writing, string capability and input-output powers of BASIC, together with the speed of execution of assembly languages. It is not atypical for 25 percent of a given program (written entirely in BASIC) to require 95 percent of the operating time. If this time-critical part can be rewritten in assembly language and efficiently linked to the remainder of the BASIC program, one can enjoy the best of both worlds."

The text begins with a general presentation of the way BASIC is organized throughout ROM and RAM. Important memory locations and the major entry and exit points of ROM subroutines are detailed. ROM utilities (registers, buffers and variables) are covered, and three chapters are dedicated to manipulation of numerical data.

Significant input/output routines are carefully explained, and a potpourri of miscellaneous Level II subroutines is also covered. If you are looking for a list of the tokens used by BASIC, they are in this book; you'll discover the easiest ways to send text to the screen, get a string of characters from the keyboard and produce text on a printer. With each and every routine that has a link to the disk system there is a boldface paragraph marked "Disk System Caution".

Part Two of *Inside Level II* applies the material from Part One, encompassing assemblers and monitors, relocating BASIC programs, VARPTR use, BASIC-assembler program linking, expansion of USR calls from one to ten, linking multiple program segments and tape load/save. The book concludes with sample composite (BASIC/assembly) programs and tape utilities to CSAVE/CLOAD composite programs at speeds higher than 500 baud. Typical program samples include preliminary comments, fully annotated source code, line by line explanations and descriptive notes. Notwithstanding the detail and careful illustration, *Inside* is by no means an easy book. We've all heard by now from the purveyors of instant solutions the problems of learning machine language. There *are* no tricks.

Although machine code and assembly language are viewed as drudgery, it's not possible to put shortcuts to work unless we have some idea of how they function. Consider that BASIC is already a kind of black box; sudden death during string sorting is just one example of that. Imagine debugging a BASIC program full of ROM subroutines—it's a shortcut to madness! Blattner and Mumford say it bluntly: "To take full advantage of the information in this book requires a knowledge of Z-80 assembly language programming."

It should be noted that Level II is undergoing some change and improvement, and newer TRS owners will enjoy better performance from their computers. Lest these alterations should cause some nervousness about the reliability of books like *Inside Level II*, let me mention that the changes being made in Level II are, for the most part, refinements and therefore will have little effect on major subroutines. Some change is made on cassette loading and saving, and a few bugs (remember POKE 16553,255?) have disappeared. The new edition of *Supermap* from Fuller Software indicates the differences between new and old Level II ROMs. ■

TRS 232 Formatter
Small Systems Software
Newbury Park, CA
$14.95

by Hugo T. Jackson

Even before I ordered the TRS 232 Printer Interface from Small System Software, I could tell from the description in the advertisement that I would not be happy with the available software needed to run it.

When compared with the rather impressive features of their TRS 232 Formatter program, it was obvious that the software sent free with the interface unit was just a bare bones program. As a cautionary measure, I order the Formatter program at the same time I ordered the equipment.

Because the supplied program makes no attempt to monitor line lengths, the first program I tried to run ended unsuccessfully. My printer output the first line of data until it reached the end of its carriage, where it stayed, typing letter over letter, until my program had kicked out what it considered to be the last character in the line. So much for the free software!

Getting There From Here

The Formatter program comes complete with a seventeen-page comprehensive manual which serves as an example of how documentation should be presented. After listing all the available options, along with a brief description of each, it details the loading procedures and memory size requirements for the program.

Having studied the manual, I reset my computer, set the memory size as required and loaded the Formatter (which is a BASIC program). When you type run, the program POKEs the machine language program into high memory

and cycles through the various options. It modifies the machine language program now in memory to reflect your choices. You can then delete the Formatter program and load or run any of your own programs.

What options are available to you? First of all, the Formatter supports nearly every baud rate I've ever heard of, and even if your printer doesn't accept one of the ten available, the manual describes how to modify the program in order to create non-standard baud rates.

If your printer requires a line feed after a carriage return, the Formatter generates one automatically. If your printer recognizes form feeds, it also generates these instead of a number of carriage returns or line feeds, to get you to the top of a new page. Needless to say, if it can execute line feeds, this saves your machine a fair amount of wear and tear.

One of the best features of the Formatter program is that it allows you to set the maximum line length. If the program you are running exceeds the maximum chosen, the Formatter automatically generates a carriage return (and line feed if needed), carrying the remainder of the text onto the next line.

The Early Line Option

Another great feature of the program is the "Line Length For Early Line Termination."

Say you have a line in your program that is eighty characters long. As your printer has only a seventy-column carriage, you have asked for a line length of seventy.

But what if the last word in the line being printed is *microcomputing*? Normally, *micr* would appear at the end of the line and the remaining letters on the following line.

By taking advantage of the early line option and requesting an "intelligent" termination at sixty characters, the Formatter program begins testing every character from sixty onwards until it finds a space, comma, colon or semi-colon. If and when it does, the program ends the line at that point, generates a carriage return, tabs inward five spaces and prints the remainder of the line. This makes for more readable listings.

If your printer requires nulls before it accepts any more characters, the Formatter sends up to 127 nulls after a carriage return before it continues sending text to the printer.

You can also set the number of lines to appear on each page, as well as the number of spaces to appear between pages. This cures many headaches, as I used to waste plenty of paper trying to prevent my printer from printing lines on the perforations of the paper.

High on the list of program features is the option which directs the Formatter to print the same information it is sending to the printer on the video monitor.

If you have disks and can't see yourself giving them to me, you should take advantage of the next option, which disables interrupts from the disk controller. If you don't, any generated interrupt destroys the baud rate timing loops in the Formatter program, and the result is garbage.

Additional Features

The Formatter program even inserts an automatic keyboard debounce routine. To top it off, there are four special features, well worth the purchase price in their own right:

To stop printing for any reason, you need only press the space bar. After that, you can print a single line at a time by repeatedly pushing the space bar. To return to uninterrupted printing, push any other key.

If the program isn't printing properly, or you typed LLIST inadvertently, you can stop the listing and return to BASIC simply by pressing the break key. No more keyboard lockup!

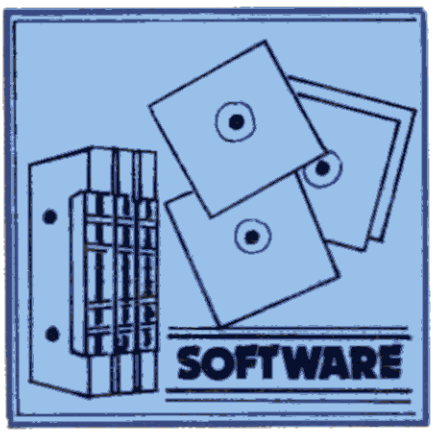
If you requested the keyboard debounce routine, you also can use the clear key to reset the lines per page counter, insuring that all your listings start at the top of the page. If you didn't request the keyboard debounce routine, you can still reset the line counter by typing LPRINT CHR$(3).

The Formatter also sends to the printer any material that is being printed on the screen (barring graphics, of course). Now I can use the TRON and TROFF functions of Level II BASIC intelligently.

While they are great features, I am sure you'll agree that the line numbers just zip by too quickly to be of any practical use. Using this feature, I can now get a permanent record of where my programs have been wandering. This option is enabled by executing LPRINT CHR$(1) and disabled by typing LPRINT CHR$(2).

Biting the Hand That Feeds

My only complaint about the Formatter program is that you cannot set a left-hand margin. Admittedly, for such a great program, this is a small point, but I do wonder why it didn't occur to them. ■



Inseq-80 and Insort-80
S&M Systems, Inc.
Marlboro, MA
TRS-80 32 and 48K Disk
$49.95

by Dennis Thurlow

As the darkness slowly parts and the mysteries that have so long enshrouded the TRS-80 disk are revealed, more and more good utility packages are finding their way to market. These packages tend to be for the software developer rather than the end-user as they require a working knowledge of disk formats and machine language routines. Inseq and Insort are no exception to the rule.

The "buy-me" on the cover of the S&M package says, "A must for anyone writing business programs." At best this is an understatement. These routines are so fundamental to the handling of disk files it's hard to understand why Microsoft didn't include them in the Disk BASIC.

Inseq uses six variables, two of which are subscripted, and four string variables, two of which are also subscripted, as well as USER functions one through six. Two invisible bytes are also added to each file.

Same Variables

Insort uses the same variables and memory that must be protected when BASIC is entered. It uses USER7 temporarily. Both programs were written to work with NEWDOS or VTOS. (If you have TRSDOS, you have to do a little rewriting.) The rewrites are well documented. The only thing that might be called a bug is that neither system will allow you to specify a drive for files you are creating, unless the file name has an extension.

The Indexed Sequential Access Method loads into high memory before BASIC is entered and the controlling BASIC program is built by another BASIC program on the disk. You may also access the functions by USER calls from your own program if you don't want to use theirs. You can read files sequentially, or refer to them by an index for reading, writing or deleting. A housekeeping function makes sure all files are closed and parameters are reset. A Data Base Utility allows you to convert already existing files to indexed sequential ones, lets you set up empty files to fill later, create a file by key from another file, and purge and reorganize files on a single disk system.

Insort allows ascending and descending sorts for ASCII or numeric fields. Any record length up to 255 is supported; keys can be any length up to full record size; sorts can be done on as many keys as memory will allow. The number generally falls between 36 and 40. A BASIC program can be generated to make the calls for you, or you can call them yourself with USER. The generated program allows you to make up to five CMD calls and/or automatically call up the next BASIC program from disk.

Sort times are admirable: 3000 files with 10 character keys can be sorted in 17 minutes, 50 files in 35 seconds. The screen constantly tells you how the sort is progressing.

Both utilities have extensive error trapping and flagging. Both are easy to use after a little practice with the demonstration files provided on the disk. For TRS-80 data file processing, both are a must! ■

Micro Music
Radio Shack
Tandy Corporation
Ft. Worth, TX
$9.95

by Allan S. Joffe

Nothing is forever. You have just played your tenth version of Star Trek and relaxed with a binary to hex program. Now the blahs have set in. The sure cure is to treat yourself to a cassette of Micro Music by Radio Shack.

This package gives you musical notes including sharps, flats and naturals over five octaves.

You control the music's tempo with rests, tremolos, triplets and staccato notes. There are also a number of voice modifying commands that change the timbre of the music.

Innovations

The documentation is adequate, but, as usual, it leaves out some surprises, which prove that Murphy will never be forced to the unemployment office while computers exist. For example, if you program a trill as (9TA8B8A8B8), it will trill nicely for about two seconds. If you accidentally leave a space between the opening parenthesis and the 9, the trill will last for about 40 seconds.

If you enter ((9TA8B8A8B8))—adding an opening parenthesis and an additional closing parenthesis—the program goes on forever unless you break it by holding down one of the arrow keys.

You can turn this bit of adversity into serendipity by trying the following program: ((9TC4R8C4R8C4R8CR8)). When you play this, the result is a Morse code V that will go on forever. This is a simple way to make a V-test tape, if that has been the missing element between you and happiness.

If you would like a snappier V, you can change the expression as follows: ((9TC8R8C8R8C8R8C2R8)), and if you want to change the tone of the V, you can insert one of the tone modifiers such as L, Y or Z between the 9 and the T.

Limitations

If your musical selections act peculiar or even weird, look out for inadvertent spaces after opening parenthesis and, the worst offender of all, an accidentally inserted closing parenthesis. This latter beast will usually cause a slow tempo, a batch of notes that you know you did not insert into the original effort.

If you have to interrupt the program while it is playing, use the right arrow key only. If you lose sight of the blinking asterisk, which is your location guide on the screen, find it by using the up arrow key only. Using the down arrow key to locate a missing cursor generally results in having to reload the system cassette.

There is another limitation worth mentioning. You do not have unlimited space to write your song. You have exactly 16 full lines of characters before you start writing over the top line. This is only destructive if you are busy transcribing a Bach fugue and forget to watch the screen!

With careful keyboard work, some study and the ability to read music, you can enjoy this fine extension of your TRS-80 and Tandy's never-ending fight to educate, instruct and, now, amuse.

The sample listing in the documentation is a real tour de force presentation of the "Flight of the Bumblebee." It's well worth the effort to key it in and then play with the tone modifiers, particularly the Z modifier. If you are old enough, it will bring back memories of the "Green Hornet."■

Micro-opoly
D. Perrin
Level IV Products, Inc.
Livonia, MI

By Alan and Nick Grassel

One of the nice things about computer games is that they usually eliminate the aggravating paper shuffling and bookkeeping that take the fun out of a game.

With Micro-opoly you get the added bonus of an honest banker. It is written in BASIC for TRS-80 Level II with at least 16K.

The single sheet of typeset instructions, will not answer all of your questions about the games operation.

As it stands, if the computer wins the roll of the dice, and goes first, you will probably find yourself shouting, "Whoa, wait a minute!" You can figure out what's happening, but it was the job of the program author to delineate this in his instructions.

The program and documentation also assume you already have an understanding of the board game Monopoly.

Your answers are entered through INKEY$. The H command is to access the information displays. The computer can either display a specific property you request or it can display a rundown of both opponents' holdings. You can use these information displays to help you plan strategy.

New zoning ordinances were passed for this game. You do not need to own all the property in a color sequence to build on any square you own, except railroad or utility.

Though the computer is limited to buying houses for only one property at a time, you can buy up to four houses or a hotel for as many squares as your bankroll will allow. This is a cash-only society. No credit allowed.

If you land on computer property, you'll have to pay rent. The computer displays "YOU LANDED ON (name)" and it flashes the property name. Then it displays "THAT COSTS YOU ($)".

If you are forced into a negative cash situation by any payments the computer requires you to sell houses or hotels until you have a positive cash balance. You also have the option of continuing to sell property until you have sufficient cash on hand to feel safe.

Programming Differences

There's a little programming quirk involved in the property purchasing which should be explained here. To add houses, enter the number you wish to add. To upgrade to a hotel, you must enter 5, although four houses plus one equals a hotel in Monopoly.

Other rule departures from the Parker Brothers' board game, Monopoly, require automatic and immediate payments when you land in jail; houses are sold back to the bank for full price (not half price); rolling doubles three times in a row will not send you to jail; you cannot sell or trade property squares; there is no Get Out of Jail Free card; and Free Parking is the repository of all funds which normally go to Poor Tax, Hospital Bills, etc. The first player to land on Free Parking gets the money as a bonus.

Purists will decry the changes, but for the most part it leads to an interesting game with a reasonable time frame. And there's no arguing about who picks up the game and puts it away.■

$MY PROPERTY-$1385
NO $$$$ PROPERTY $ #HOUSES

*YOUR PROPERTY-$1500
NO **** PROPERTY * #HOUSES

2	MED.AV	0	4	BALTIC	0
6	READ #	3 OWNED	9	VT.AVE	0
7	ORIENT	0	12	CHARLY	0
10	-CONN-	0	13	ELEC	10 X DICES
14	STATES	0	15	VA.AVE	0
16	PENN #	3 OWNED	17	JAMES-	0
19	-TENN-	0	22	KY.AVE	0
20	NY.AVE	0	26	B&O #	1 OWNED
24	IND.AV	0	27	ATLANT	0
25	ILL.AV	0	28	VENTNR	0
30	MARVIN	0	29	WATER	10 X DICES
32	PACIF-	1	35	PA.AVE	0
33	NC.AVE	0	38	PARKPL	0
36	S&L #	3 OWNED	40	BWALK	0

Fig. 1. Specific Property Display

Model 440 Paper Tiger
Integral Data Systems, Inc.
Natick, MA
$995 standard printer
$1094 with graphic option

by James H. Sheats

When it became obvious that my printer was inadequate, I started looking for a new one. Naturally, I wanted as good a printer as possible. Dealers whom I asked declared that Integral Data Systems' Brighter Writers had performed satisfactorily with only minor breakdowns. They recommended the IDS Model 440 Paper Tiger.

Removable Case

The Tiger has an attractive, removable molded plastic case. It can be removed by twisting four knurled retainer nuts.

All of the controls are reached through cuts in the plastic case or are mounted on the metal back plate of the printer. A fuse socket, a 115 V-230V selector switch and the main power switch are all found on the back plate. There are no unguarded openings, so even a clumsy fellow like me can reach behind the printer for the switch with relative safety.

In the top of the case on the right are the offline/online switch and a form feed/line feed switch. The printer must be offline while the "self-test" is administered or any form feed adjustments are made. On the left side of the tractor are the formset/test switch and two banks of seven DIP switches.

The right position of the formset/test switch is the self-test pattern. As long as the switch is held to the right, the printer prints full line lengths of the 96 ASCII characters at the selected print density. This part of the self-test also helps the user align the paper.

Part of the self-test is supposed to occur automatically upon power up. Spontaneously printed characters presumably indicate a malfunction.

Both paper tractors are continuously adjustable, so odd-sized paper widths can be handled, as well as the more common widths, up to a maximum of nine and one-half inches. A paper roll holder is available as an option.

Two sets of DIP switches control the remaining adjustable features of the printer. Though plainly marked, they are small and close together, and you might easily move more than one switch at a time.

Switches one and two on the left bank control the adjustable print sizes—8.3, 10, 12 and 16.5 characters per inch. The print line is eight inches long, making the line length 66, 80, 96 or 132 characters, respectively. The printer ignores changes in these two DIP switches, unless the main power switch is turned off. It is probably a good idea to turn off the main power switch while reconfiguring any of the DIP switches.

Switch three selects an eight-line per inch or a six-line per inch vertical spacing. Letters and normal printing can be at six lines per inch, while manuscripts and other matter requiring double spacing can be printed at eight lines per inch to make them less wasteful of paper.

Switch four enables or disables a one-inch skip at form boundaries. I have found this feature useful for program listings.

Switch five enables or disables an automatic line feed with carriage returns. If a TRS-80 program was originally written for a printer without an automatic line feed, the necessary line feeds were probably incorporated in the program, and this switch must be disabled. The Electric Pencil definitely requires that this switch be off.

The Paper Tiger utilizes a 7×7 dot matrix with upper and lowercase letters. Its speed varies with character size, but is in the 50–100 characters per second range. This is fast enough for most hobby use.

No Mechanical Failures

Overall mechanical performance of my Paper Tiger is excellent. I have had no mechanical breakdowns that required service.

I have had certain software problems operating the Tiger in conjunction with my TRS-80, though. I found out that TAB characters do not work past the 63rd character in a printer line. There are, of course, programming alternatives to the TAB command.

Once my Tiger hung up on a program listing. The BASIC line that caused the trouble was a long, multi-statement one. I simply broke the line into several lines of BASIC and had no more trouble of this nature. ■

Maillist
DAR Sales
Sacramento, CA 95812
$39.95

by Chris Brown
***80* Staff**

There you sit with a substantial investment in computer hardware, wondering how to make it pay its share of the rent. Everyone tells you it's easy to get rich using computers, but nobody you know is close to breaking even. Then, along comes Richard Alva from DAR sales.

He says you can make some bucks with your machine by providing a service sorely needed in your community. This service involves performing a rather simple, but dreary task, generating mailing labels.

Ideal computer stuff, eh?

With DAR software and Alva's business tutelage, you will be able to establish and maintain mail lists for local businesses and organization. The result: instant money for you and your computer. Sound too good to be true? Maybe it is, since the success is in the selling, and Richard Alva leaves that to you.

Alva Has a Better Idea

Richard Alva has had a better idea. Instead of going through the hassle of selling his software directly to users, he has gone a step further and produced a business package for computerists. It is up to them to find and then sell their users.

The package is called Maillist and comes with software documentation and 100 promotional letters. Alva's intention is that these will be used to drum up interest in the service. He even includes an instruction manual detailing how to set up your business.

The author has calculated the costs involved in supplying the service and offers a suggested price list: ten cents for entering a three-line name and address; add three cents if a special code is used; add five cents if remarks are entered, etc.

You say you haven't got a printer? Well, don't worry. Richard Alva has thought of that too. Just copy your data tapes or disks and send them to DAR. For a nominal fee (three cents per label), Alva will do the printing for you. The same is true for sorting. Normal program sort priority is by zip code but, if necessary, DAR will sort your data in other formats. Again, for a nominal fee.

The Software

The program cassette supplied has two program dumps. The first is a 32K, disk-based version of Maillist, the second is a 16K tape-based version. The disk version requires only one drive and most of the additional memory required for the disk version is taken up by operating system instructions.

The Maillist program creates three files on a disk. Each file is capable of holding up to 125 names and addresses, and entries are coded to the disks. Each disk can hold 375 entries. The user manual provides detailed instructions for opening, changing and deleting entries in the files. The step-by-step intructions seem clear enough for the most novice of operators.

A file sort time in the standard zip code format takes only 15 minutes, according to Alva.

If you think you have a knack for selling and want to get a sideline going, DAR's Maillist package can give you a start. It comes with a 30-day, money back guarantee, if not fully satisfied, and a promise from Richard Alva that you will earn at least ten dollars an hour for your trouble. If he can sell you, maybe you can sell too. ■

TRS-80* OWNERS:

- **Let the computer write your "Basic" program for you!**
- **Draw pictures, animated figures, data forms!**
- **Create a library of display forms!**
- **Produce "Commercial" gradesoftware!**

The **Magic Cursor** is a Revolutionary Family of Products which provides a dramatic new method of reproducing drawings and displays that you create on your screen. It makes both simple displays and complex interactive data input forms. It stores a "BASIC PROGRAM" on disk (or tape) ready for you to execute alone or as a subroutine. It produces screens in both standard or wide screen.

It is available for any level 2, 16K or larger system with tape or disk. An optional version is now available which creates an assembly language program.

Be sure to pick out the system that fits your present needs and order it today. You may upgrade your original copy by paying the difference and a moderate service charge.

MAGIC CURSOR PROGRAMS

THE MAGIC CURSOR allows you to easily create screens (including graphics) on your video. A powerful command then generates the BASIC instructions to recreate the screen. For the first time, a program for automatic generation of video display forms. (16K Tape or Disk) **$24.95**

THE MAGIC CURSOR I additionally makes sophisticated Data Entry and Display easy. With Magic Cursor I you define the Data Entry or Display fields directly on your screen. The definition commands generate the BASIC instructions to implement the Data Entry and Display. The Magic Cursor I has commands which move, center, and duplicate blocks of graphical or alpha/numeric displays. You can even justify text. (16K Tape Only) **$79.95**

THE MAGIC CURSOR II adds the power to write animated games easily in BASIC. The Magic Cursor II allows you to reload previous screens either from memory or from Disk. You can then modify them and store either the modified screen or only the changes. (32K Disk Only) **$99.95**

THE MAGIC CURSOR III will be available soon for the new Model II Computer (32K One or more Disk) **$149.95**

THE MAGIC CURSOR IV provides the features of Magic Cursor II but stores an assembly language program. (32K Disk Only) **$99.95**

WRITE FOR OUR COMPLETE SOFTWARE CATALOG!!

CUSTOM COMPUTER CENTER, INC.™

For ordering or information write:
P.O. Box 58042 / Houston, Texas 77058
Attn: Jim Martens
or call: (713) 474-2484

✓7

NEW RELEASES FROM CCC!!!

Now available for the first time!!!!!
A Monitor/Trace program with versions for both Model I and Model II.

Trace-80™

TRACE-80 lets you observe the inner working of a machine language program. It allows you to run a machine language program in slow motion and watch the screen. You can stop execution at any time and examine the current instruction mnemonic and all register contents. You can execute your program and watch each instruction mnemonic and register contents list to the screen in place of normal screen display.

If you have a printer, **TRACE-80** allows you to execute your program in slow motion and watch the screen while your printer simultaneously prints the machine code being executed, the memory location and the instruction mnemonic along with the current register contents.

You can execute a machine language program in slow motion, freeze the action, examine and/or change memory, examine and/or change register contents and then continue the slow motion. You can speed up past common routines and slow down to examine other routines in detail or operate in single step mode.

TRACE-80 allows you to trace ROM as well as RAM because instructions are emulated in a special execution buffer.

FEATURES:
- ★ For both beginner and advanced programmer.
- ★ More than 20 commands.
- ★ Trace-80 is written in machine language.
- ★ Traces both ROM and RAM.
- ★ Level II or Mod I Disk operation.
- ★ Model II Disk version available.
- ★ Optionally prints only "Transfer and Control" instructions.
- ★ Full speed, slow speed or freeze execution modes.
- ★ Memory can be displayed/modified.
- ★ Register contents can be displayed/modified.
- ★ Hex, ASCII and mnemonic display modes.
- ★ Abreviated or full printer format.
- ★ Serial printer output if desired.
- ★ Option of normal screen display, memory display, trace display or clear screen.
- ★ Learn assembly language programming as well as machine coding by watching actual code execution and see assembly language mnemonic.

PARTIAL LIST OF COMMANDS: Load disk file, Trace, Slow Motion Execution, Full Speed Execution, Freeze Action, Single Instruction Execution, Examine and/or Display memory, Examine and/or Display Register Contents, Enable/Disable Screen, Enable/Disable Printer, Ascii or Hex Display, Full Screen Memory Display, Line Printer Commands, etc.

TRACE-80/MOD-I (for Level II or DOS operation) **$29.95**
Supplied on tape with 3 versions (16K, 32K or 48K). Complete with instructions.
TRACE-80/MOD-II **$49.95**

The Restauranteur's Consultant
by Ty Halderman

This food and beverage management tool dramatically reduces the human factors in food cost analysis. Those tasks required to effectively operate any restaurant or food service business. Over a half dozen reports give uniform, accurate and up to the minute information for profitability. And, handling of daily cost changes requires only minutes per week instead of hours, because the Consultant makes all the necessary conversions from your case prices. Reports include:

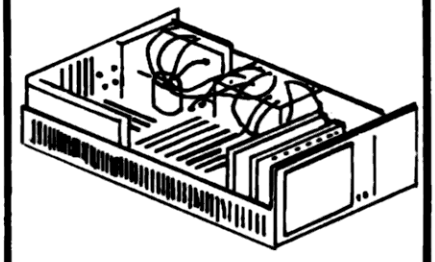
1. Menu Recipes.
2. Ingredient Listing.
3. Supplier Master File.
4. Complete Listing of Food Cost Analysis.
5. Food Cost Summary.
6. Input Data Sheets.
7. Batch Update.

Mod 1, 32K, 2 Disk, Printer or Mod 2 ..... **$750.00**

*Trademark of Radio Shack, a Tandy Co.

✓Reader Service—see page 226

HARDWARE



DB-9500 Line Printer
Anadex, Inc.
Chatsworth, CA 91311
$1650 Model I Compatible

by Edward E. Umlor

The DB-9500 is one of the better designs among dot matrix printers on the market. Some of its features include:

- I/O Parallel, serial and current loop interfaces, all built in;
- Type: two fonts available (normal size 9×9 and condensed size 7×9) for 10, 12 and 13.3 characters per inch, and six or eight lines per inch (double width format is also available in both fonts);
- Speed: from 150 CPS to 200 CPS, depending on type selected;
- Paper: 1.75 to 16.875 inch, edge punched, single weight from 15 to 100, or multipart thickness of 0.018 inches maximum;
- Ribbon: enclosed cartridge;
- Graphics: individual dot addressable in 7×1 format (each character is one dot wide and 0-7 dots high.

All parameters (except I/O format) are software selectable as well as switch selectable. The DB-9500, selected for parallel (Centronics format) operation, was hooked up to our TRS-80 word-processing system and fired up. The printer performed correctly on the first run. The printer ready line prevents data from going to the printer prematurely so that all characters are printed. We had the 2K optional RAM installed for a total of 2.7K FIFO buffer.

The print quality is very good in both fonts and is crisp and clear. The lever with detent stops at the left side of the carriage controls the distance of the print head from the platen. Set the lever for maximum separation to load the paper more easily and then set it back for the correct impact pressure. This is a much better arrangement than the fixed gap method used by most printers.

Its operation is almost flawless. We were running a mail list on our MSI computer, with the perforation skip set at one inch. On each new page, the top two lines did not have the correct separation. This might have been caused by paper drag as the paper came out of its box. When half an inch skip setting was used, there wasn't a problem with paging. ■

Microline-80
Okidata
Mount Laurel, NJ
$800

by Gary L. Osburn

I was recently faced with the $64 question (or in this case, the $1000 question): Out of the dozens of printers available for use with the TRS-80, which one would be the best for me?

I was infected with the micro-bug about two years ago and found that the only treatment was massive doses of Z-80! When I decided to take the plunge, the lack of ready cash meant that the habit had to be self-supporting. I started a small consulting service for developing custom software and needed a printer that emulated any of several printers that my clients were using. The printer had to look like it printed 40, 80 or 132 columns.

The Okidata

At the National Computer Conference in Anaheim I found an Okidata display. In one corner of the display was a sleek little box sitting on a pedestal spitting out fact sheets at 80cps. I thought it particularly effective to have the printer printing its own specifications.

The printer is impressively small. Its vital statistics are 13.4 inches wide × 9.4 inches deep × 4 inches high. Apparently the trick to producing such a small printer is the low mass print head. Energy is stored in tension members waiting to fire extremely hard print pins on demand. This translates into low heat and longer life. The company is so sure of the longer life that it guarantees the print head for 2,000,000 characters.

The printer produces 162 lines per minute (80 column format) and can print either six or eight lines to the inch. This can be a real paper saver!

Extremely clear characters are printed in a 9×7 dot matrix. The character set is the ASCII standard 96 characters (upper and lowercase) and features block graphics! This capability allows for all sorts of clever printing possibilities. (See Table 1.)

The printer comes standard with friction and pin feed (ten-inch pins). A tractor drive mechanism is available for $140 and simply snaps into place when it is needed. It can be removed when it is not.

You can use the inexpensive roll paper for most of your work, and still have the tractor when needed. Incidentally, standard Teletype paper available at most office supply shops is a lot cheaper than what Radio Shack sells.

Speaking of supplies, the Microline-80 uses a standard two-inch typewriter ribbon that you can pick up almost anywhere for less than two bucks!

Using the Microline-80 is a real dream. There were no installation problems whatsoever, the ribbon connector sent with the unit fit perfectly, and the connection pins required no rewiring. The quality of the print really helps when looking for that hard to find bug. Though the lowercase letters do not have below the line descenders, the print is definitely of letter quality. Consider the Microline-80. ■

```
This is a sample of 80 column printing:
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
abcdefghijklmnopqrstuvwxyz

This is a sample of 132 column printing:
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!#$%&'()*<>?,./:=-
abcdefghijklmnopqrstuvwxyz

THIS PRINTING IS DONE AT 6 LINES PER INCH
THIS PRINTING IS DONE AT 6 LINES PER INCH
THIS PRINTING IS DONE AT 6 LINES PER INCH

THIS PRINTING IS DONE AT 8 LINES PER INCH
THIS PRINTING IS DONE AT 8 LINES PER INCH
THIS PRINTING IS DONE AT 8 LINES PER INCH
THIS PRINTING IS DONE AT 8 LINES PER INCH

GRAPHICS AT 80 COLUMNS
```

Microline-80 Output (Actual Size)

THE ASSEMBLY LINE

by William Barden, Jr.

"The scheme for creating one large program with EDTASM/ Apparat is to keep adding source code lines as required."

This month we'll be discussing editor/ assemblers for the Model I and II and more about SET/RESET.

Radio Shack's Disk Editor/Assembler, a macro assembler that produces relocatable modules, differs from EDTASM, the cassette based Radio Shack Assembler and from the Apparat version of EDTASM, in the following areas:

1. It produces relocatable object modules that must be "linked-loaded."
2. It allows some limited macro capability.
3. It has some format differences in the pseudo-op area.
4. It allows some additional pseudo operations relating to titling, page formatting, and so forth.

The assembler does basically the same job as the Apparat assembler, taking a source assembly program in standard Z-80 mnemonics and translating it into object code, operating with disk files for source and object. A source program for the disk assembler will look virtually identical to one for EDTASM or the Apparat assembler. The difference lies mainly in how it goes about producing the source code.

Relocatable Modules vs. One Program

The scheme for creating one large program with EDTASM/Apparat is to keep adding source code lines as required. Within the source code, you can structure separate program modules. A large assembly language program would probably have many separate subroutines with defined inputs and outputs, and several levels of routines, as shown in Fig. 1.

All the code, however, would be within one source file. There would be no problem referencing a label in one part of the program from another part, as the assembler would have built up a symbol table of all labels used in the program.

This scheme of one huge program is fine except for two gremlins, memory size and divisibility.

As RAM is being used to hold the source code and the symbol table, there is a limit to the size of the program that can be assembled. This limit is a function of the number of lines of source code, size of the lines, and number of labels used. I've reached the limit in about 1000 source lines with a lot of comments and a lot of symbols (it's terrible programming practice to have "JP $+257"!). Memory size, therefore, may be a problem for large programs.

The second gremlin, divisibility, requires some explanation. How do you divide a program when you run out of memory and you have a huge program that is crisscrossed with references? What about a large programming task that must be split up among several programmers? How do they write code that can be merged together efficiently?

Assemblers that produced relocatable object modules were developed in the early days of programming to get rid of the twin gremlins of limited memory size and divisibility. The Disk Editor/Assembler allows a large program to be broken up into as many modules as desired.

In this type of assembler, each source code module is assembled after the edit to produce a relocatable object code module. Why relocatable? Obviously, it would be difficult to assign absolute addresses for each object module, as the sizes are variable. Each module is relocated at load time by automatically adding a relocation bias to the addresses and other relocatable data types.

Intermodule communication is handled by EXTernal and ENTRY pseudo-ops. If a label is declared as an ENTRY, other modules may reference the label, provided they have declared a corresponding EXT for the name of the label. These labels are referred to as global, because they are accessible to all modules, rather than just locally, inside of one module.

To see how this process of assembling, loading and linking works, see Program Listing 1, which shows a huge program of three modules.

Module one is the main module, typically the driver program. In this case MAIN looks for a keypress of 0-7, prints the key and loops back for the next keypress. MAIN references two other modules, KEY and PRINT. KEY is a subroutine to detect a keypress and PRINT prints the value of the keypress at the screen center.

Note the EXT for KEY and PRINT in MAIN and the corresponding ENTRY pseudo-ops in KEY and PRINT. Of course, all modules may have both ENTRYs and EXTs, depending on the references.

The commands to the loader to load MAIN, KEY, and PRINT from disk as object modules and to write the subsequent core image out as PROG is given in Program Listing 2. PROG can be loaded and executed as a single CMD type file from disk.

Wholly Macro! Look at This Feature!

A second feature of the Disk Editor/ Assembler is its ability to define and use macros. A macro in its simplest form is nothing more than a specified set of instructions that is spewed out at assembly time when the macro name is invoked. Suppose, for example, that we used the sequence of instructions

```
LD      HL,BUFFR
LD      DE,DCB
LD      B,0
CALL    4424H
```

several times in program to OPEN a TRSDOS disk file. Rather than writing the four instructions for assembly each time we open a disk file, we can define the instructions as a macro called OPEN. The MACRO pseudo-op defines the label as a macro name, and the code between the MACRO and ENDM defines the body of the macro. Having defined the macro at the beginning of the assembly language source code (Program Listing 3), we can now invoke the macro (OPEN), automatically generating the four instructions, at any point in the source code.

Not only does the assembler allow us to define sets of instructions as macros, it allows us to use general arguments as parameters for macro calls. In Listing 3, we don't always want to use BUFFR as the buffer; we might want to

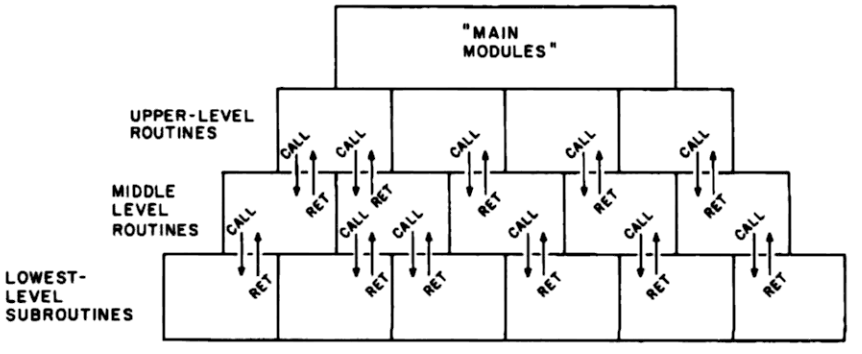


Fig. 1. Typical Assembly Language Program Hierarchy

```
                        00100   ; MAIN DRIVER CALLS KEY AND PRINT
                        00110           EXT     KEY,PRINT
0000'  3E 20            00120   START:  LD      A,' '           ;BLANK
0002'  11 FFFF          00130           LD      DE,-1           ;-1 FOR DECREMENT
0005'  21 03FF          00140           LD      HL,1023         ;FOR 1024 CHAR POS
0008'  DD 21 3C00       00150           LD      IX,3C00H        ;START OF SCREEN
000C'  DD 77 00         00160   LOOP1:  LD      (IX),A          ;CLEAR SCREEN
000F'  DD 23            00170           INC     IX              ;BUMP POINTER
0011'  19               00180           ADD     HL,DE           ;DECREMENT COUNT
0012'  38 F8            00190           JR      C,LOOP1         ;GO IF MORE
0014'  CD 0000*         00200   LOOP2:  CALL    KEY             ;LOOK FOR KEY
0017'  CD 0000*         00210           CALL    PRINT           ;DISPLAY
001A'  18 F8            00220           JR      LOOP2           ;LOOP DA LOOP
                        00230           END

                        00100   ; KEY SUBROUTINE LOOKS FOR KEY PRESS OF 0-7
                        00110           ENTRY   KEY
0000'  3A 3801          00120   KEY:    LD      A,(3801H)       ;GET ROW
0003'  B7               00125           OR      A               ;TEST
0004'  28 FA            00130           JR      Z,KEY           ;GO IF NONE
0006'  C9               00140           RET                     ;RETURN
                        00150           END

                        00100   ; PRINT SUBROUTINE DISPLAYS KEY ON SCREEN
                        00110           ENTRY   PRINT
0000'  06 FF            00120   PRINT:  LD      B,-1            ;COUNT
0002'  0F               00130   LOOP:   RRCA                    ;SHIFT OUT A
0003'  04               00140           INC     B               ;BUMP COUNT
0004'  30 FC            00150           JR      NC,LOOP         ;GO IF NO CARRY
0006'  78               00160           LD      A,B             ;COUNT
0007'  C6 30            00170           ADD     A,30H           ;CONVERT TO ASCII
0009'  32 3E28          00180           LD      (3E28H),A       ;PUT ON SCREEN
000C'  C9               00190           RET                     ;RETURN
                        00200           END
```

Program Listing 1. Typical Relocatable Modules

```
0000                    00100   ; SAMPLE OF A SIMPLE MACRO
0100                    00110   BUFFR   EQU     0000H
                        00120   DCB     EQU     0000H+256
                        00130   OPEN    MACRO                   <-- MACRO DEFINITION
                        00140           LD      HL,BUFFR
                        00150           LD      DE,DCB
                        00160           LD      B,0
                        00170           CALL    4424H
                        00180           ENDM
0000'  3E 05            00190   TEST:   LD      A,5             ;MEANINGLESS
                        00200           OPEN                    <-- MACRO CALL
0002'  21 0000                  LD      HL,BUFFR
0005'  11 0100                  LD      DE,DCB                  <-- MACRO EXPANSION
0008'  06 00                    LD      B,0
000A'  CD 4424                  CALL    4424H
                        00210           END
```

Program Listing 3. Simple Macro.

```
DOS READY
L80                     (Loads Loader)
* -P: 8000              (Sets load location to 8000H)
*MAIN, KEY, PRINT       (Loads MAIN, KEY, PRINT
DATA 8000, 802F         in that order)
DATA 8000 802F          (Program boundaries)
*TEST-N,-E              (File TEST/CMD, end load)
(0000 802F)
DOS READY
```

Program Listing 2. Typical Load Sequence

use BUFFR1 or BUFFR2. By defining dummy arguments in the macro definition, a macro will utilize a given set of arguments everytime it's invoked. Program Listing 4 shows three arguments, BUFFER, DCB and LRL, as dummies in the macro definition. When the OPEN macro is invoked, the arguments specified are then substituted for the dummy as shown.

Macros can be used to simplify calls to subroutines or system functions—as they are on large computers—or to automatically generate a set of in-line code, or even to define a special assembly-time interpretive language.

Format Differences

The Disk Editor/Assembler has some minor format differences from EDTASM. Labels on assembly source lines must be suffixed by a colon. The pseudo-ops for defining bytes, words, storage and strings are either DEFB, DEFW, DEFS and DEFM or an alternate (8080) set of DB, DW, DS and DC. One of the nicest differences in these pseudo-ops are that multiple arguments can be used as in DB 2,5,45,6,77.

The editor uses similar, but not identical, commands to the BASIC/EDTASM editor; the commands are not quite as powerful as in EDTASM. An edit is performed on a disk file. At the completion of an edit, the modified file is written out to disk as a new file name. The old file name cannot be used, necessitating a KILL followed by a RENAME to complete the edit of a source file.

Some additional pseudo-ops in the Disk Editor/Assembler allow listing format control and conditional assembly. Those in the first group are such commands as TITLE, SUBTTL (subtitle), PAGE and .COMMENT. Listing can be selectively controlled for various parts of the program. Conditional assembly is controlled by pseudo-ops such as IFT (if true) and ENDIF. Other functions, such as repeat (REPT) code, are also permitted. All in all, the commands incorporated into the Disk Editor/Assembler are similar to the commands one sees in assemblers on most minicomputer systems.

Which Assembler is Best?

Which assembler, EDTASM/Apparat or the Radio Shack Disk Editor/Assembler, is "best"? Unfortunately, an assembly language programmer is almost forced to choose one or the other, because of minor format differences between the two. This is really a lamentable condition; it's a shame that colon in labels wasn't exorcised.

The Disk Assembler got short shrift in an earlier column. After using it extensively however, I've formed the following opinions:

"Which assembler, . . . is 'best'? Unfortunately, . . . a programmer is almost forced to choose one or the other, because of minor format differences. . .

- If the TRS-80 is used in a programming department in a commercial company, the best choice of assemblers is probably the Disk. It provides all the features of EDTASM/Apparat and with its Modular flexibility can maintain and develop large program packages.
- If you do a great deal of assembly language programming or want to use the best tools available, then the Disk Editor/Assembler again is probably most useful.
- If you are learning assembly language programming or do most of your assembly language programming in short code segments, then stick with EDTASM/Apparat or the new Microsoft version of EDTASM, EDTASM-PLUS (cassette based).

These, are my personal opinions, but I'd be interested in hearing yours. (By the way, I'd also be interested in hearing your opinions about the content of this column: Should it be at a lower level, higher level, oriented towards ROM calls, more tutorial, more product oriented? Please let me know.)

Model II Assemblers

Disregarding assemblers that run in a CP/M (Control Program for Microcomputers) envir-

```
                        00100   ; SAMPLE OF A SIMPLE MACRO WITH ARGUMENTS
                        00110   OPEN    MACRO   BUFFER,DCB,LRL
                        00120           LD      HL,BUFFER
                        00130           LD      DE,DCB
                        00140           LD      B,LRL
                        00150           CALL    4424H
                        00160           ENDM
                        00170   TEST:   OPEN    8000H,8000H+256,0
0000'  21 8000                          LD      HL,8000H
0003'  11 8100                          LD      DE,8000H+256
0006'  06 00                            LD      B,0
0008'  CD 4424                          CALL    4424H
                        00180           END
```

MACRO DEFINITION WITH DUMMY ARGUMENTS

MACRO CALL WITH REAL ARGUMENTS

MACRO EXPANSION WITH REAL ARGUMENTS

Program Listing 4. Macro with Arguments.

onment on the Model II, there are at least three others which will run on that computer.

The first is the Radio Shack Macro Assembler. This is a rewrite of Microsoft's Disk Assembler for the Model I and has all of the same features that we discussed in the earlier part of this article. The cost is $199.

The second is from Galactic Software, Ltd. EDAS 4.0 has a text editor identical to the BASIC editor. Though I have not used the product, according to its specifications, all object output can go directly to a disk file or to memory and it executes all TRSDOS commands. Its cost is $229.

The third is MACASM from Racet Computes. This is an interesting product, as it is a EDTASM-PLUS modified for the Model II. EDTASM-PLUS is a another Microsoft Assembler, and contains an editor, assembler and debug program all in one package. The editor is basically the same editor which is in EDTASM, but has several new commands, such as Move Block and Copy Block.

The assembler is a macro assembler that can be assembled directly to memory. The Z-BUG portion is a symbolic debugger that performs virtually any useful debug function you can imagine, including referencing of locations by symbolic assembly name. Switching between editing, assembling and debugging is instantaneous, as all three are resident at one time.

MACASM is part of a Mod II Development package that includes SUPERZAP, MACASM and Disassembler and costs $125.

The Continuing Saga of SET, RESET

I received a long letter from Jerald J. Kovacic and William Sit, who collaborated on a fast multiply (see previous column). The multiply was fast, but the mail delivery was not. In Jerald's letter he describes the ROM call for POINT, SET and RESET, which should be of interest to many readers.

Basically, all three calls require that there is an ASCII character string in memory of the form (X,Y) where X and Y are legitimate column and row numbers of 0-127 and 0-47, respectively. Caution! The ASCII string must not contain invalid X or Y values and must have correct syntax, otherwise the following routines will go to a BASIC error processing routine, and . . . goodbye assembly language!

For POINT, the call is:

(HL) = pointer to ASCII message of (X,Y) minus one
CALL 0132H
(Return with location 4121H = 0 if point off, 0FFH if point on;
((HL)) = pointer to first non-blank following the string)

For SET and RESET, the calls are:

(HL) = pointer to ASCII message of (X,Y)
CALL 135H for SET
CALL 0138H for RESET
(Return with ((HL)) = pointer to first non-blank following the string)

How do these CALLs compare to the SET/RESET routine given in an earlier column? If you recall, we set or reset about 2500 points per second. It appears that these ROM routines allow us to operate on about 140 points per second, not to mention the overhead of setting up a string in ASCII of X and Y values. This checks with a tight BASIC SET/RESET loop figure of 122 pixels per second that has somewhat more overhead. The test program used to time it is given in Program Listing 5.

In past columns I've been reluctant to discuss ROM calls. There are some good reasons for this. Firstly, I question how many modular, well defined routines there are in BASIC. Secondly, problems, such as the one above, can branch out in to cloud-cuckooland. Finally, I think one learns more by writing his own assembly language routines. Comments? ■

```
                        00100   ; TEST PROGRAM FOR ROM CALL FOR SET/RESET
0000'  DD 21 0000       00110           LD      IX,0            ;ZERO COUNTER
0004'  21 0014'         00120   LOOP:   LD      HL,MSG          ;STRING ADDRESS
0007'  CD 0135          00130           CALL    0135H           ;SET
000A'  21 0014'         00140           LD      HL,MSG          ;MESSAGE ADDRESS
000D'  CD 0138          00150           CALL    0138H           ;RESET
0010'  DD 23            00160           INC     IX              ;BUMP COUNT
0012'  18 F0            00170           JR      LOOP            ;LOOP
0014'  28 36 30 2C      00180   MSG:    DB      '(60,30)'       ;X=60, Y=30
0018'  33 30 29
                        00190           END
```

Program Listing 5. Test Program for ROM SET/RESET

INSIDE 80

from page 10

impressed. I seriously considered—quietly—buying one myself . . . didn't want the boss to know . . . but the price/need ratio was still only mildly attractive. The deciding factor in my case might have been one of judgment.

I'm not a digital or computer technician, but I am experienced enough in electronics to understand Murphy's law and its effect on computer hardware compatibility. If you buy two items from the same supplier, you at least have one company you look to for proper interfacing. If they come from different suppliers, about all you can reasonably expect is that each company assure proper operation of their particular item.

Radio Shack has a limited supply of technical types, and we hope it is now obvious what they've been doing since we introduced Model II. From the 1977 introduction of Model I, we've said many times, that we can't offer specialized hardware help or custom programming.

We see other folks' ads for some pretty impressive-looking TRS-80 add-ons and software. We haven't had hands on with much of it though, and so we really don't know how to make it play with TRS-80s. Selfishly (we all get that way sometimes), I have to suggest that it might be more reasonable to ask for help from the folks who asked you to buy that product for your TRS-80. Of course, we will support our advertised applications on our equipment.

99-cent Catalog

Contrary even to what my friend Wayne Green would have you believe, we don't try to hide our competition from our customers—especially software (his August Remarks). He overlooked our Software Sourcebook, a 99-cent catalog of over 1,000 "non-Radio Shack" TRS-80 programs. Remember, Wayne, you paid us $10.00 each to list seventeen of your own Instant Software programs.

Wayne and I have known each other for close to 20 years now, and I know he won't take offense at a friendly jab, so I'll also chide him a bit for mis-reading his Tandy Annual Report. Our total payroll (not just for our administrative "bureauracy") for operating 7,353 stores worldwide, including six headquarters locations in Fiscal 1979 was about 18 percent of our total expenses rather that the "almost 50 percent" he suggested.

In fact, compared to the top ten computer companies, our sales per employee were second only to IBM.

Be all of that as it may, we members of the bureaucracy will continue to bring you new products, both hardware and software, to the best of our ability. And again, I assure you that your comments and suggestions will be met with open arms. As soon as we find the remainder of our management staff—those we're buying for the other 30 percent of our expenses, Wayne—maybe we can offer individual replies. ■

SAFOR

Sales Analyst and Forecaster

with Graphic Display

Here is a new program from Software Etc. . . that is invaluable to any businessman. SAFOR is a time series analysis and forecasting program that will produce presentation quality graphs on your own printer. SAFOR will handle up to ten years of monthly data, provide a comprehensive analysis of past patterns and make both long and short-range forecasts.

SAFOR uses a classical time series decomposition model to provide both tabular and graphic presentations of Original Data, 12 Month Forecasts, Business Cycle, Growth Cycle, Seasonal Pattern and 12 Month Moving Average.

SAFOR can handle any type of data series measured on a monthly basis where no value is zero or less. This data may be in any unit of measure, however SAFOR does not adjust dollar figures for inflation. The trend is calculated in the same units as actual sales, while other factors are treated as multipliers or indexes. Because the irregular component is unpredictable, SAFOR ignores it.

SAFOR is designed for ease of use, data entry is particularly straightforward, every user action is prompted. The program features extensive editing, updating, data storage and error detection routines.

SAFOR allows for varying levels of expertise in the techniques of analysis, beginner to expert. For the beginner, SAFOR contains standard default operations to help prepare routine analysis and forecasts. For the expert, key assumptions in the program are readily modified without any programming.

SAFOR will run on your TRS-80® Level II with an expansion interface, 32K of RAM memory, a disk drive, TRSDOS® and optionally a 132 column printer.

Each program comes complete with a two year set of demonstration data. Order yours now! A $200.00 value at this Introductory Price of only **$79.95** Good through December 31, 1980 Only.

✓42

Software Etc. . .
1839 Chamberlain Drive,
Carrollton, Texas 75007.
Phone Orders: (214) 492-0515

Demand a Demonstration from your local dealer, or write for a brochure of our complete line of fine software.

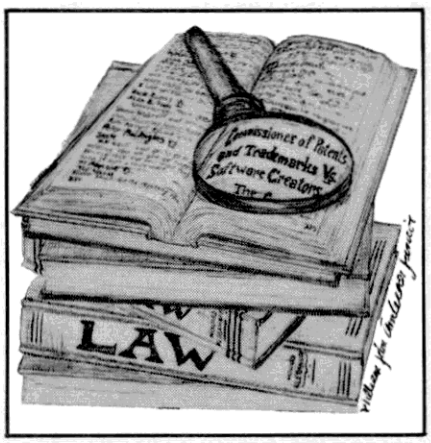
Reader Service—see page 226

80 NEWS

edited by Nancy Robertson

"Machines. . . are defined by their function—and it is the software that determines the function of a general purpose computer."

Are Computer Programs Patentable?



While the industry as a whole clamors for H.R. 6934 (a bill that offers copyright protection for software), a patent granted July 1, 1980 has been overlooked. Patent 4,210,961 for a sorting system was granted to Whitlow Computer Systems, Inc. of Englewood Cliffs, NJ. The grant was made for "a method. . . of sorting data. . . utilizing a digital computer and at least one random access device. . . ."

Although the patent is specifically for "a method," the sort system is part of Whitlow's software product Syncsort for IBM OS computers. The decision is a coup for all advocates of software patenting.

Currently the U.S. Patent Office is not accepting applications for software or firmware. The Whitlow application was filed prior to the present policy. Thomas Lynch, legal counselor for the Patent Office, explains that judicial decisions have indicated that patents do not apply to computer programs.

Strong Contingent

However, advocates of software patenting counter that the Patent Office is shirking its duty. This small, but strong contingent is fighting for legal recognition of software programs as inventions, which would make software eligible for patent incentives.

The legality of the Patent Office's stand will be considered by the U.S. Supreme Court when it reconvenes this month. Two programming patent cases will be heard: *Diamond, Commissioner of Patents and Trademarks,* v. *Bradley and Franklin*; and *Diamond* v. *Diehr and Lutton*. The first case is a patent appeal for a firmware ROM chip. Diehr and Lutton's appeal is for a software patent.

While the legal debate is fanned in Congress and the courts, it's worth noting some distinctions between patents and copyrights. According to *Webster's New Collegiate Dictionary*, a copyright is "the exclusive right to reproduce, publish, and sell the matter and form of a literary, musical, or artistic work." A patent is "a writing securing to an inventor for a term of years the exclusive right to make, use, or sell his invention." Before slapping the dictionary shut, an invention is "a device, contrivance, or process originated after study and experiment."

Michael Keplinger, chief counsel to the special government Commission on New Technological Uses, points out that a copyright is not a government grant. "It exists in something, if it is a work, from the moment it is created."

Generally, a work is considered to be copyrighted from the moment the pen leaves the page, without any formality. To be fully protected by copyright benefits, Keplinger suggests the copyright be registered with the Library of Congress. Two thousand copyrights were registered for software in 1978 and 1979.

Keplinger also talks about "the critical difference" between a patent and copyright. "A copyright is anti-rip-off protection. Copyrighted software can't be copied with only trivial changes—but you could take a close look at somebody else's software and take the main ideas to make your own program. Anyone else could create a program based on the same algorithm and copyright. But a patent is given for a process implemented in a program. After one is granted, writing another program with the same algorithm would be a patent infringement."

Major Objections

This is exactly the objection the patent office has to programming patents. According to Thomas Lynch, "The basic rule of thumb is that anything can be patented if it is new and an improvement of what has been done before. But with programming (both firmware and software), you're talking about an old or existing machine programmed to do something different. A patent of a program is a patent for an idea rather than for the development of a new structure."

However, businessmen such as Whitlow's president Aso Tavitian have "never considered copyright as worthwhile protection." Whitlow's Syncsort was jointly developed by several people in the small software company early in the '70s. A patent application was filed at that time. "The feeling was—and it was shared by all of us—that we had an invention. . . . My feeling is that to think about copyright as equivalent is 'way out'. . . . The purpose of a patent is to encourage innovation. It's like a reward: For 15 years we have a legal monopoly on this sort."

"The purpose of a patent is to encourage innovation. It's like a reward: For 15 years we have a legal monopoly on this sort."

Whitlow's attorney Morton C. Jacobs explains that the Syncsort patent is "not a program listing patent, as such, but a patent on a sorting system built with software." Jacobs has filed one of the several "friend of the court" briefs that the Supreme Court will consider in respect to the patent cases it is hearing this fall. His argument is particularly germane to microcomputers and other general purpose computers.

"Machines," Jacobs says, "are defined by their function—and it is the software that determines the function of a general purpose computer." He argues that "a new machine is created every time a new piece of software is entered."

In the brief submitted to the Supreme Court, Jacob states that "To remove the sorted program from the general-purpose computer is to remove its unique rule of action as a machine. This would be the same as untying the wires interconnecting the circuits of a hardware-

continues to page 45

Computer Merchandisers Hurt by Mail Fraud

No arrests have yet been made in Chicago area towns following a rash of thefts which has left several computer supply houses around the country holding bad checks totalling nearly $140,000. The scheme of theft by deception first came to light earlier this summer when complaints of bum checks began trickling into the Riverside Police Department from a number of out-of-state computer houses who had shipped merchandise C.O.D. to a Riverside address.

Outwardly, the customer appeared legitimate enough, bearing all the trappings of a bona fide business. According to Lt. Dokupil, chief of detectives with the Riverside PD, the goods were ordered by CMI, Inc., a phony company which maintained an office at 3340 S. Harlem Ave. An individual using the name of Thomas Janson (or "Jansen") used the premises as a base of operations from which he placed orders around the country for a variety of merchandise.

The same general procedure was used in all of the thefts. An order would be phoned in by CMI to a computer firm for a sizeable amount of software or hardware. At the time of delivery, the driver would accept the CMI check as payment for the goods, per standard procedure. Because the company had carefully created the illusion that it was in fact an established firm with a permanent business address, nothing out of the ordinary was noticed—until checks started bouncing.

By the time the complaints reached the local police, the suspects had already fled and an investigation of the Harlem Ave. address revealed only a hurriedly vacated office. After Riverside, Janson and his accomplice apparrently set up shop in nearby towns, crow-hopping to a new location whenever things got hot.

One of the dealers who got stung was a Texas supplier who received a phone order from CMI for two hundred eight-inch and five-inch diskettes which the caller wanted shipped out as a rush order. The order was sent out C.O.D. the next day, and when the UPS driver delivered, payment of $600 was effected by personal check. When the dealer received the check his suspicions were aroused by the absence of a company name and number on the face of the check; the scrawled signature of "Thomas Janson" was just barely discernible at the bottom of the check.

The dealer had little choice but to go ahead and deposit the check, and hope for the best. His suspicions were confirmed several days later when the check was in fact returned: The Illinois bank account upon which it had been written was no longer in existence. Neither CMI nor Janson were listed in the phone book. It was at that point that the dealer contacted the police in Riverside. At about the same time, similar reports began to reach the police from other dealers around the nation such as one in California who had been bilked out of $3000 worth of hardware (color boards) in exactly the same manner.

The same scam was repeated numerous times from many different locations in the area surrounding Riverside. In each case CMI would abscond just before the police could close in for an arrest.

Most recently, the operation was headquartered in Morton Grove, IL. It is in this town that police feel they have compiled the most concrete evidence to date with felony warrants being issued for the arrest of Janson and his accomplice.

According to Det. Redman of the Morton Grove Police Department, a picture of the prime suspect has been distributed to law enforcement officials involved in the case in other locals.

Assistance has also been sought from the FBI due to the interstate nature of the crimes and federal charges will be leveled if and when any apprehensions are made. Among these is a "theft by wire" charge which is brought in instances where a telephone is used as an instrument of larceny.

Lt. Dokupil advises all dealers who are defaulted to go directly to their local police with as much information as they can pull together. Dealers are urged to be doubly cautious when doing business with new clients, specifically, on C.O.D. orders. They are being admonished to accept only cash, money orders (cashiers' checks), or established charge cards.

One dealer, still smarting from his recent ripoff at the hands of CMI said, "We hate to do it, but now have to be much stricter with payment procedures; this (loss) is cash right out of our pockets." ■

By Paul Quinn
80 Staff

Are Computer Programs Patentable?

continued from page 44

program computer, or disassembling the wheels, gears and levers of a mechanically programmed machine..."

In reference to the possibility of copyrighting programs, Jacobs' brief states, "Copyright protects particular expression, not machine structure." Jacobs feels definitions of the words machine and mechanical are at the crux of the patent question. While the Patent Office argues that programs are not mechanical and are not machine structures, Jacobs points to a finding of fact in the recent *Data Cash Systems, Inc.*, v. *JS&A Group, Inc., et al* case: "The computer program (stored in a computer) is a mechanical device which is engaged in the computer to become an essential part of the mechanical process."

Whether programming is artistic creation or creative engineering will be open to aesthetic and intellectual debate as long as computing continues to advance. But the Supreme Court will rule either for or against programming patents this fall. If the court rules in favor of patents, programmers will still not have universal protection. Patents are not easy to come by in any field. As Thomas Lynch explained, to be patented an invention must be "new and an improvement of what has been done." To be copyrighted, a work needs only to be completed—and artistic. ■

By Nancy Robertson
80 Staff

Future Home Computing

"The Home of the Future," the second annual Yankee Group symposium on home information utility, will be held Oct. 14-15 in Palo Alto, CA, and Oct. 21-22 in New York City. It will outline and discuss how access to computing power will be provided, who the suppliers will be, and how this trend will affect all users of electronic information processing.

According to the Yankee Group, the Home of the Future will be part of an information bus. Data bases such as The Source, CompuServe, and Knight-Ridder will be easily accessible.

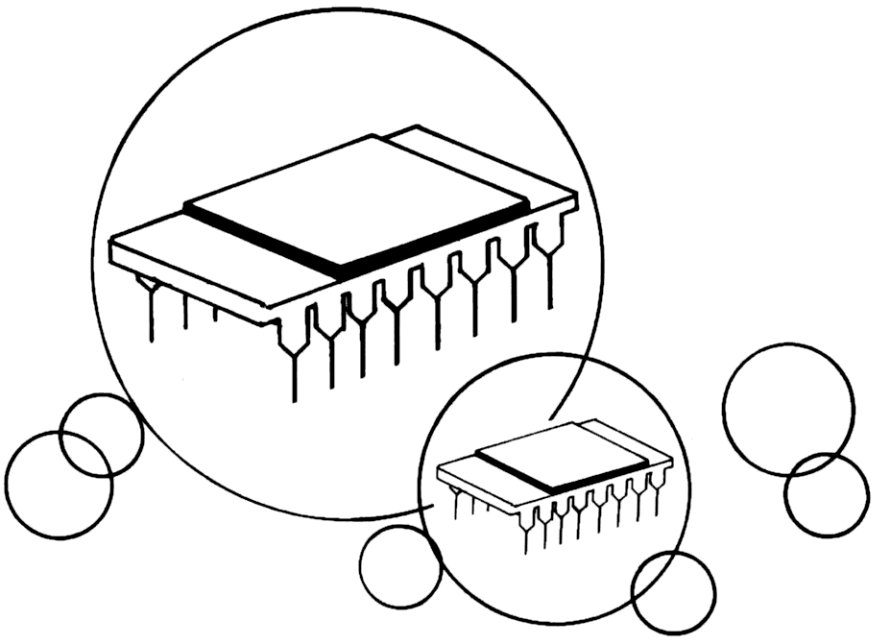
But information of and by itself cannot exist in the marketplace. It must be carried by either telephone lines, coaxial cable, or the cost must be partially covered by other vendors. The Yankee Group believes that broadband communications must come to the Home of the Future.

The seminar will demonstrate how information access could be provided by either the telephone industry or the cable systems, and what the implications of either's actions will be.

Speakers at the symposium include: Ted Turner, Cable News Network; Gus Hauser, Chairman of Warner/Amex (owners of Qube, Columbus, Ohio); George Minot, Compuserve; Tom Harnish, OCLC (which is putting information data bases on line in 4,000 libraries); Irving Kahn, Broadband Communications; Jack Taub, The Source; and Howard Anderson, The Yankee Group.

For further information and registration, contact Marjorie Sugarman, The Yankee Group, P.O. Box 43, Harvard Square, Cambridge, MA 02138. ■

A Slow Road to Bubble Memories



The mere mention of bubble memory devices gets a rise out of most jaded floppy disk jockeys. Unfortunately, bubbles may not prove to be the memory panacea most expect them to be for some time.

The name of the game is mass storage and, for the microcomputerist, the options are few. Mass storage refers to the ability to save relatively large amounts of data in a nonvolatile medium. The three most popular mass storage methods currently in use are cassettes, disks (floppy and otherwise) and Winchester systems.

"It is no wonder that hobbyists are always looking for a better method of mass storage: one that is reliable, fast. . . and. . . cheap."

Each method has drawbacks. Cassettes are notoriously unreliable and are extremely slow. Also, individual cassette storage capacity is limited.

Disks are more reliable, have much faster access time, but are expensive. They also require an operating system which usually resides in RAM and uses up about 12K of memory in a TRS-80.

Winchester systems are expensive, and often a pain in the neck in terms of software support and compatibility.

It is no wonder that hobbyists are always looking for a better method of mass storage: one that is reliable, fast, easily maintained and, last but certainly not least, cheap. Enter rumours of the bubble.

Bubble memories are the latest development in the technology of mass storage. They are made of neither solid core material nor semiconductor material. Instead, they are composed of a microscopic film of magnetic material that is deposited on a nonmagnetic substrate or base. The magic of bubble memories is their ability to create zones of magnetization at discrete points throughout the magnetic film. These tiny magnetic points are opposite in polarity from the rest of the magnetic film, and each of these magnetic bubble zones represents one bit of information.

Bubbles are nonvolatile, that is they retain their magnetic polarity indefinitely without requiring external power, and are so small that millions can reside in the space of one IC. Space requirements are so minimal in fact, that a standard DIP (dual in-line package) bubble chip has 1M bit of storage capacity (100K bytes). That's roughly the equivalent of a single density, 40 track, 5¼-inch floppy disk.

Bubble memory systems do have some disadvantages. The control and support circuitry necessary to access the bubbles is considerable, often requiring an individual control chip for each bubble chip. In addition, since bubble addressing is loop configured and semi-serial, data transfer rates are not particularly fast. A Texas Instruments' bubble system has an average transfer rate of 85K bits per second compared with a TRSDOS formatted disk transfer rate of 100K bits per second. Manufacturers are working on these problems now, and advances on both fronts should be forthcoming.

The three major bubble memory manufacturers today are Texas Instruments, Intel Corp., and Rockwell Int. Each offers bubble devices on both component and board levels, but none are seriously considering entering the personal computer market with their systems at this time.

George Riggs, a spokesman for Rockwell, explains that he does not see a small system market for bubble devices developing "for at least three or four years." He added that "when the cost is lowered to around 15 millicents per bit in bubble systems, the hobbyist will find them an alternative to other systems." This inevitable cost reduction will take time, however. Rockwell does have a bubble system available now. Their 256K bit board can be bought for $1800.

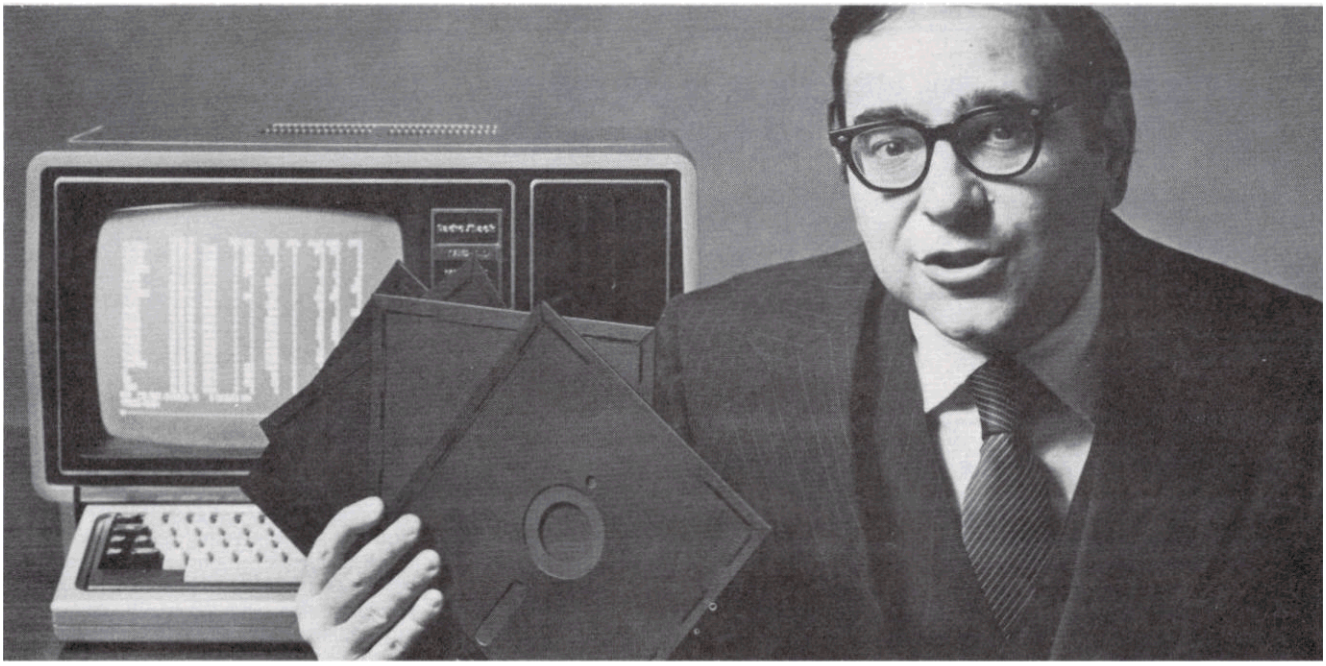
"For the present, bubble memories do not seem to be a practical alternative for the vast majority of small system computerists."

Intel of Santa Clara, CA, markets a board-level bubble system in kit form. Their 7110-1 Magnetic Memory board comes complete with all control and support circuitry, and sells for around $2000. Judy Kochanowski of Technical Marketing at Intel feels that although no plans are afoot to crash the hobbyist marketplace in the near future, their 7110-1 kit is "competitively priced" when compared with the current cost of a four-disk system for micros.

For the present, bubble memories do not seem to be a practical alternative for the vast majority of small system computerists. The problems related to support and control in both hardware and software are beyond the range of all but fanatical uses to solve. In addition, the cost per Mbyte of storage in bubble devices has not yet fallen low enough to justify their use.

Finally, none of the major manufacturers in the bubble industry take the small system user very seriously. When, and if, these manufacturers sense the existence of a market for their bubble devices in the microcomputer field, rapid developments in small system bubble devices can be expected. Until that time, micro users will continue to CLOAD, disk dump, watch and wait. ■

By Chris Brown
80 Staff



I made the TRS-80* into a serious computer. Now I've made the Model II into a spectacular one.

I'm Irwin Taranto, and I've helped almost a thousand businesses get their first computers up and running.

I've done it primarily with the TRS-80, because it's a really elegant piece of hardware. Given the right programs, it can do substantially the same work as the traditional minicomputers that cost four times as much.

I proved it with four on-line, interactive programs adapted from the genuine Osborne & Associates systems, originally designed for the $30,000 Wang computer. Then I added two of my own and made them all work on a $4000 TRS-80.

Now I've done the same thing for the new TRS-80 Model II. It's an $8000 computer that works twice as fast and has four times the memory—up to two million characters.

My new systems are fully documented, and because I'm working with a much more powerful computer, they're a night-and-day advance over the Model I programs. They'll turn your Model II into a complete business computer, set up and ready to go.

THE TRS-80 MODEL II PROGRAMS

General Ledger/Cash Journal: handles up to 7000 transactions on 500 different user-defined accounts. It keeps track of them by month, quarter and year, makes comparisons to the prior year, and does departmentalization.

Accounts Payable/Purchase Order: generates the purchase order and posts the item to payables when the goods are received. Invoice-linked, it calculates and prints checks and aged ledger reports and links fully to the general ledger.

Accounts Receivable/Invoicing: keeps track of billed and unbilled invoices, open and closed items, aging and service charge calculation. It prints statements, links to the general ledger, and can work within either an invoice-linked or balance-forward accounting system.

Payroll/Job Costing: computes regular, overtime and piecework pay, keeps employee files, figures taxes and deductions, prints checks, journal, 941-A and W-2 forms, and breaks out individual job costs.

*A trademark of the Tandy Corporation.

When I say set up and ready to go, I mean just that. If you're not quite sure on that point, call the number below and we'll give you the names of some of the people who've already bought all over the world. Call them up and hear what they have to say.

These Model II programs are completely custom-tailored, which explains their $249.95 price. Before we'll send you a disk, you have to fill out a detailed questionnaire that tells us your precise business requirements. Then we send you the disk, all the instructions you need, and my phone number. If you call, we answer all your questions. If your questions are tough enough, I'll talk to you personally.

Because that way I'll make sure that Model II of yours turns into a spectacular computer, just like I promised.

☐ Please send me the custom questionnaires for the following $249.95 Model II programs:

- ☐ General Ledger/Cash Journal
- ☐ Accounts Payable/Purchase Order
- ☐ Accounts Receivable/Invoicing
- ☐ Payroll/Job Costing

☐ Please send me information on the TRS-80 Model I programs at $99.95 each

☐ Please send me information on other Taranto business programs

Your name _____

Company name _____

Address _____

City/State/Zip _____

✓45

Taranto
& ASSOCIATES, INC.

P.O. Box 6073, 4136 Redwood Hwy., San Rafael CA 94903 • (415) 472-2670

✓Reader Service—see page 226

NEW PRODUCTS

The Micrographics Index

The National Micrographics Assoc. (NMA) Resource Center has released the latest 1980 computer-output microfiche edition of the Micrographics Index and Special Interest Package #15 entitled, *Service Bureau vs. In-House Systems*, a collection of current articles that are part of the Resource Center's inventory.

Over 250 new entries have been added since the Index was last published in January 1980. The Micrographics Index is a catalog of the over 4,000 items contained in the NMA Resource Center. It provides direct, comprehensive access to the largest collection of micrographics in the world.

The Index includes information on micrographic applications, technical processes, case histories, standards, research reports, equipment evaluations, directories, "how to" guides, state-of-the-art reports, market studies and industry surveys. Journal, author, keyword and subject indices provide access to entries listed in the Index. Reprints of most items may be ordered from NMA in hardcopy or microfiche for a small fee.

Service Bureau vs. In-House Systems is a collection of articles that overviews the considerations in deciding to use a service bureau or to implement an in-house micrographic operation. Topics include evaluating and selecting a service bureau, justifying in-house COM and case histories of COM and source-document in-house operations.

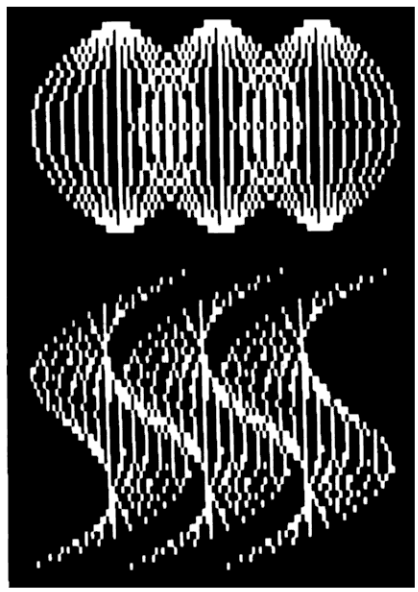
Service Bureau Versus In-House Systems is available in papercopy for $25 and in microfiche for $15. The Index can be purchased for $35. Both are available from NMA Publication Sales, 8719 Colesville Rd., Silver Spring, MD 20910.

Reader Service ✓ 173

Advanced Graphics Course

Datagraphics, P.O. Box 566, Dept. G, Union Station, Endicott, NY 13760, has a new program, Advanced Graphics Mini-instruction Course, Volume 1, Curves. The program is the first in a series of projects on graphics applications programming techniques.

Volume 1 uses a new algorithm for the old problem of plotting curves: None of the displays that are generated use advanced math, such as sines and cosines; a simple arithmetic progression-regression technique is used instead. Graphics are executed on video within five to 20 seconds.



Graphics Designed with Datagraphics Mini-instruction course.

The program begins with a simple explanation of FOR-NEXT loops and line numbers, continues with amplitude equations, regressions and progressions. There is also a program included for designing computer art.

Supplied on tape for 16K Level II or 4K Level I, Advanced Graphics Mini-instruction course sells for $20.55.

Reader Service ✓ 160

Macro Library Adds Nearly 80 New Mnemonics

Stoneware Microcomputer Products, 1930 Fourth St., San Rafael, CA 94901, is selling an extended instruction set macro library to work in conjunction with Microsoft's Macro-80, Digital Research's MAC and CDL/TDL Macro I and Macro II.

A macroassembler is defined in the *Microcomputer Dictionary and Guide*, Matrix Publishers, Inc., as something that "simplifies coding when similar sections of code are used repeatedly, but variations preclude the use of conventional subroutine techniques."

Stoneware's new library expands the Z-80/8080 instruction set with nearly 80 new mnemonics. They enhance the hardware instruction set by creating pseudo instructions which are reconstructions of the exsisting Z-80/8080 instruction set. The assembler mixes the new instructions with the existing set.

The library has been designed to be compatible with current software. It sells for $109.95 for CP/M systems with eight-inch single density disks, and for $79.95 for Model I TRSDOS on 5¼ inch disk systems.

Reader Service ✓ 169

Do-it-yourself Interfacing

A five-page booklet is available that provides instructions, schematics, a parts list and software driver listing for do-it-yourself interfacing between a TRS-80 and an RS-232 printer. According to the booklet, which costs $4.95, the interface can be constructed for less than $2 in parts.

The booklet, which was not named in the company announcement, is sold by Fobel Enterprises, 552 E. El Morado, Ontario, CA 91764.

Reader Service ✓ 180

Business Data Base Plus Statistical Package

Charles Mann & Assoc. has a new programmable Business Data Base System for the TRS-80 and a new Statistical Package. Business Data Base System allows the user to define and build data bases for such purposes as inventory control, general ledger accounting, accounts receivable and accounts payable. The fields may be manipulated with a math formula accumulator to format reports and present status displays.

The system is compatible with TRSDOS, NEWDOS and 3.0 DOS. The programs are auto linked and called automatically as needed. Business Data Base System needs at least 32K RAM with 48K recommended, and a single disk drive. Multiple disk systems are supported under user control.

The Business Data Base System is available for $89.95 from Charles Mann & Assoc., 7594 San Remo Trail, Yucca Valley, CA 92284.

The Statistical Package includes Scientific Data Management System and a number of

ADVENTURERS! RISE TO NEW DEPTHS!

At Last, 3 Dimensions!

Deathmaze 5000 and Labyrinth are the first in a new breed of adventure. Instead of wandering through the English language, typing GO EAST or GO WEST, you move through a colossal maze represented on the screen three-dimensionally. Hallways recede into infinity or come to dead-ends. Doors open to right and left. Pits open in floor and ceiling. As you encounter objects, monsters, and mayhem, one or two word commands may be used. The command set is extensive and sophisticated. The proper commands allow the solution of problems and the manipulation of objects. The improper choice of words could spell the end. . . .

MACHINE LANGUAGE SOPHISTICATION

Deathmaze 5000 and Labyrinth are written in machine language. They are both incredibly *fast*. All the features expected of great adventures are built in, including SAVE GAME and a blinking cursor. All versions include relocation modules for use with disk systems.

Deathmaze 5000 places you on the top floor of a five story building. Each floor is a maze of twisting passageways. Floors are connected by elevators and open pits. You have but one goal. **ESCAPE ALIVE!** Where is the only door out of this nightmare? Monsters, bats, mad dogs, hunger, and many more horrors plague your every step as you struggle to escape the most complex adventure ever written.

TRS-80 Level II 16K cassette $12.95
APPLE II or APPLE II PLUS 32K cassette $12.95

Labyrinth places you in a maze of gigantic proportions. But you are not alone! A minotaur searches for you, seeking a grisly meal. You must find weapons, spells, and treasures. You must deal with ghosts and cave gnomes. You *must* avoid the minotaur until the moment is right for the final battle.

TRS-80 Level II 16K cassette $12.95
APPLE II or APPLE II PLUS 32K cassette $12.95

Med Systems Software

P.O. Box 2574 Chapel Hill, NC 27514
(919) 933-1990

★ ★ ★ CONTEST ★ ★ ★

REWARD!

This man escaped a fate worse than death. He was the first, but we hope not the last. From his condition, you would never realize that he designed **Deathmaze 5000**. Those few others who survive may send their correct solutions to us. On December 31, a drawing will be held. Six intrepid adventurers will win their choice of three Med Systems programs and a shirt silk-screened with the above logo and the words "I survived Deathmaze 5000". Only the correct solutions are eligible. All judgements final. Please enclose a SASE for return of solutions or notification of correctness. All winners will be contacted directly.

SATISFACTION GUARANTEED!

All Med Systems Software products come with a 14 day moneyback guarantee. If for any reason you are not satisfied, return your order within 14 days for a prompt and cheerful refund.

ORDERING INFORMATION

Orders are processed within two working days. Mastercard and Visa card holders please remember to include the expiration date. We pay all postage and handling within the U.S., Canada, and U.S. territories. European orders please include $2.00 for air post.

Ask about our other adventures. These include Samurai, Reality Ends, Bureaucracy, and The Human Adventure.

Please send the following 3-D adventures:

☐ Deathmaze ($12.95) $_____

☐ Labyrinth ($12.95) $_____

TOTAL $_____

☐ Please send your catalog of programs and products, as well as details of the Deathmaze contest.

Name _____

Address _____

City _____ State _____ Zip _____

Computer:
☐ TRS-80 16K LII ☐ APPLE II or APPLE II PLUS 32K

☐ Mastercard ☐ VISA ☐ check

MC or VISA # _____

Expiration Date _____

✓Reader Service—see page 226

other programs for curve fitting, probability, general statistics, distribution mathematics and test statistics. The programs allow for curve fitting of data using linear, exponential, logarithmic, or power relationships. A built-in data base is included to produce a set of working examples to help users learn statistical principles.

The system requires 32K of RAM and at least one disk drive. It can handle up to five 100-by-1 matrices containing raw data, grouped data or frequency arrays. The package sells for $89.95.

Reader Service ↙ 171

Single-key Entries for DOS

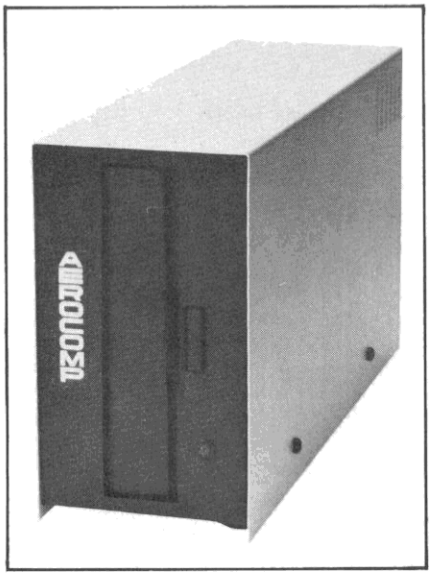
Mediamix has a program for TRS-80 disk users called Super Directory. While in the DOS command mode, the user enters D; the computer then asks for the drive number and an indexed directory of that drive's disk is displayed along with a menu of command modes. A single keystroke will put the user in RUN, KILL, FREE or PRINT.

Super Directory is a machine language program for TRSDOS or NEWDOS users. It is sold for $9.95 by Mediamix, P.O. Box 8775, Universal City, CA 91608.

Reader Service ↙ 172

Faster Drives Are for Sale

Aerocomp, Inc., P.O. Box 24829, Dallas, TX 75224, has a new line of disk drives with the MPI drive acting as the center of the system. With an access time of five milliseconds, the company claims that MPI drives are the fastest in the field, and offer the most accurate disk positioning in the industry. Centering is accurate within 0.0008 inches.



Aerocomp's New Disk Drive

The Aerocomp drives' power consumption is low at six Watts standby, 12 Watts operating. All write-protect and index sensing is accomplished optically. They also allow users to flip disks to utilize both sides.

No pricing information was included with the company's product announcement.

Reader Service ↙ 167

Double-density Storage Boost

Percom Data Co., Inc., 211 N. Kirby, Garland, TX 75042, has begun production of a double-density disk-controller adapter for the TRS-80 Model I. The adapter is called Doubler. According to Percom's calculations, using Doubler an 80 can store and format up to 354K on a five-inch disk, compared to the 256K of a standard eight-inch floppy.

Doubler is sold with DBLDOS, which is a TRSDOS-compatible double-density operating system, and with a utility to convert single-density files and programs to double-density format. The complete system sells for $219.95 from Percom.

Reader Service ↙ 170

T/Maker Report Generator

A software tool, combining a tabular report generator with word processing, is available from Lifeboat Assoc. The T/Maker system provides easy analysis and presentation of numerical data and text copy used in financial modeling and report preparation. Typical T/Maker applications include sales projections, profitability studies, balance sheets, estimates, price sheets, etc.

The system includes a full screen editor. A macro command allows any series of keystrokes to be saved and executed with one keystroke. Text insert, delete, global search and replace, and block move are all supported by the editor. Computation for rows and columns includes: standard arithmetic; percents; exponents; common transcendental functions; averages; maxima; minima; projections, etc.

The T/Maker requires a 48K CP/M system and C-BASIC-2. It costs $275 from Lifeboat Assoc., 1651 Third Ave., New York, NY 10028.

Reader Service ↙ 177

Eight-inch Floppy Drives

Parasitic Engineering's Maxi-disk eight-inch floppy disk drives are now compatible with the TRS-80 Model II. Used with the Model II, Maxi-disk drives are functionally identical to Radio Shack expansion drives. No software or hardware changes are needed.

Each drive is contained in its own cabinet. Additional drives are simply plugged in, so that a drive can be removed for service without disturbing any other drives on the system.

Maxi-disk Drives

The drives cost $845, plus $60 for the three-drive cable which is needed for connection. They are sold by Parasitic Engineering, Inc., 1101 Ninth Ave., Oakland, CA 94606.

Reader Service ↙ 163

Video-oriented Text Editor

Southeastern's Textan is a machine-language text editor designed to operate on 16K machines. It is a video-oriented editor designed for BASIC programmers. Textan reads program tapes written in Level II BASIC and returns to BASIC with the program fully loaded when editing is completed.

This text editor has 32 command functions and 26 reserved-word keys. The command functions provided include: previous screen, next screen; auto line numbering; block delete; display free memory, etc. The reserved word keys will automatically enter many standard BASIC commands.

The package costs $40, plus $7.50 for the manual, from Southeastern Software, 512 Conway Lane, Birmingham, AL 35210.

Reader Service ↙ 179

Model II General Ledger

GL is a general ledger system for the TRS-80 Model II. It requires TRSDOS 1.2, a 132-column printer, a dual disk system, and 64K memory. It is part of a larger accounting system which includes A/R, A/P, and Payroll. Summary transactions from these packages are accepted automatically by GL.

The GL package produces departmental and summary income statements showing current and year-to-date amounts, percentages by category, and comparative data with the general ledger one year ago. The chart of accounts contains a five-digit account number; account description; and current, year-to-date, and budget balances. Users specify account

type, master/sub account code and balance sheet column code.

The major outputs are trial balance, balance sheet, income statement and department income statements. The major programs are Master File Maintenance/List, Transactions/Enter/Register/List, Account Status/List, Generate Financial Reports, Accounting Transactions Transfer, End-of-period Processor and System Initialization. ISAM is used for fast random key and sequential access.

GL sells for $129 from Micro Architect Inc., 96 Dothan St., Arlington, MA 02174.

Reader Service ↙ 165

Horse Handicapping Program

A horse race handicapping package for the TRS-80 and Apple home computers is being sold by the 3G Co. The company has gathered and stored data from a vast number of races, and analyzed which attributes contribute to a horse's performance in a race, either positively or negatively.

The package consists of a guide on how to use the "Daily Racing Form" to obtain the ten factors needed for each horse, a sample form to simplify the data gathering, a cassette that computes the odds for the current race, a program listing for use with other types of computers, and tips on how to use odds when wagering.

The package costs $19.95 from 3G Co., Rt. 3, Box 28A, Gaston, OR 97119.

Reader Service ↙ 164

Low Cost Dot-matrix Printer

DIP, Inc., is selling a low cost Data Impact Printer, the model DIP-81. It is priced at $499. The model DIP-81 is a dot-matrix impact printer, designed for continous duty cycle.

The model DIP-81 features 7-by-7 or expanded 14-by-7 matrix printing, an upper/lower case character set, 100 characters per second in a bi-directional print-out and ribbon cartridge loading. It uses ordinary bond paper in sheets, roll or fanfold form.

With full 96 character ASCII set, it is capable of both upper and lowercase printing at both 40 and 80 characters per line. Operator control includes power, select/deselect, line feed, top of form and self test. A Centronics compatible parallel interface is standard. Serial RS-232C or 20mA current loop is optional.

Contact DIP, Inc., 121 Beach St., Boston, MA 02111 for more information.

Reader Service ↙ 185

BASIC to FORTRAN

The Management, Box 111, Aledo, TX 76008, has a programmer utility called FORTRANslator. It is designed to aid in the translation of TRS-80 Disk BASIC Model I programs to TRS-80 FORTRAN.

FORTRANslator converts BASIC into the structured READ, WRITE, FORMAT constructs. It also translates BASIC key-words and procedures such as IF-THEN-ELSE into correct style. FORTRAN indentation and spacing, C lines, DO loops and other conventions are produced. A program can be created and debugged in BASIC, then translated to compiled FORTRAN.

This machine-language program will run on a 32K machine with at least one disk drive. A printer is recommended. FORTRANslator is priced at $29.95 and is supplied on a Model I data disk.

Reader Service ↙ 178

Land Surveying Applications

Four land surveying programs for TRS-80 have been developed by Disco-tech. They are Field Note Data Reduction, Coordinate Geometry, Stadia Reduction and Horizontal Curve Staking. They are included in Disco-tech's Survey 80 package.

The Survey 80 package has been developed and field-tested by a team of practicing land surveyors. The four programs allow users to choose various methods of solving technical problems. Fifteen-digit accuracy is built in where appropriate, and ouput can be displayed on the screen and/or printed out.

No computer knowledge is presupposed. Steps are presented logically and lucidly on the screen. Programs are supported by manuals which guide the user step-by-step through hardware installation, data entry, computation, and output. The manuals are reinforced by practical examples and appendices which treat common problems and care of magnetic media.

Survey 80 is available in a TRS-80 Model I version, and by November will be offered in a CP/M version.

Field Note Data Reduction costs $250, Coordinate Geometry costs $350, Stadia Reduction costs $175 and Horizontal Curve Staking costs $95. The programs are sold by Disco-tech, a division of Morton Technologies, Inc., P.O. Box 11129, Santa Rosa, CA 95406.

Reader Service ↙ 162

Video Football with Strategy

Acorn Software Products, Inc. has another new game—Pigskin, a football strategy game for the Model I Level II.

Two players can compete against each other, or one player may challenge the program in one of five levels of difficulty. Any game in progress can be saved.

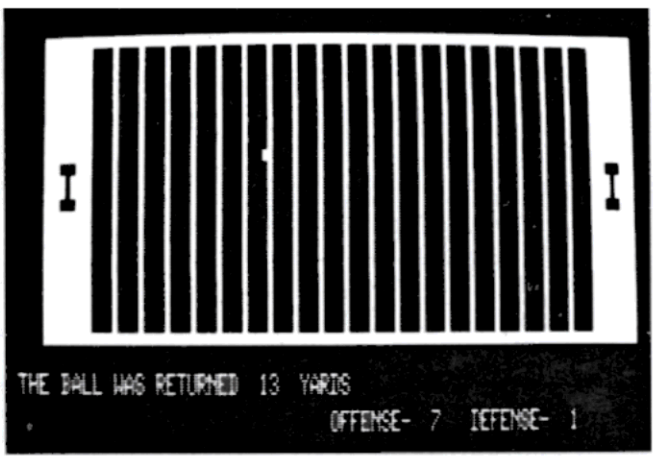
Pigskin's graphic display of the field shows ball movement and statistics as players employ their skills. Strategy involves the use of ten offensive plays and six defensive positions.

The game is priced at $9.95 on cassette, or $15.95 on disk from Acorn Software, Inc., 634 North Carolina Ave., S.E., Washington, D.C. 20003.

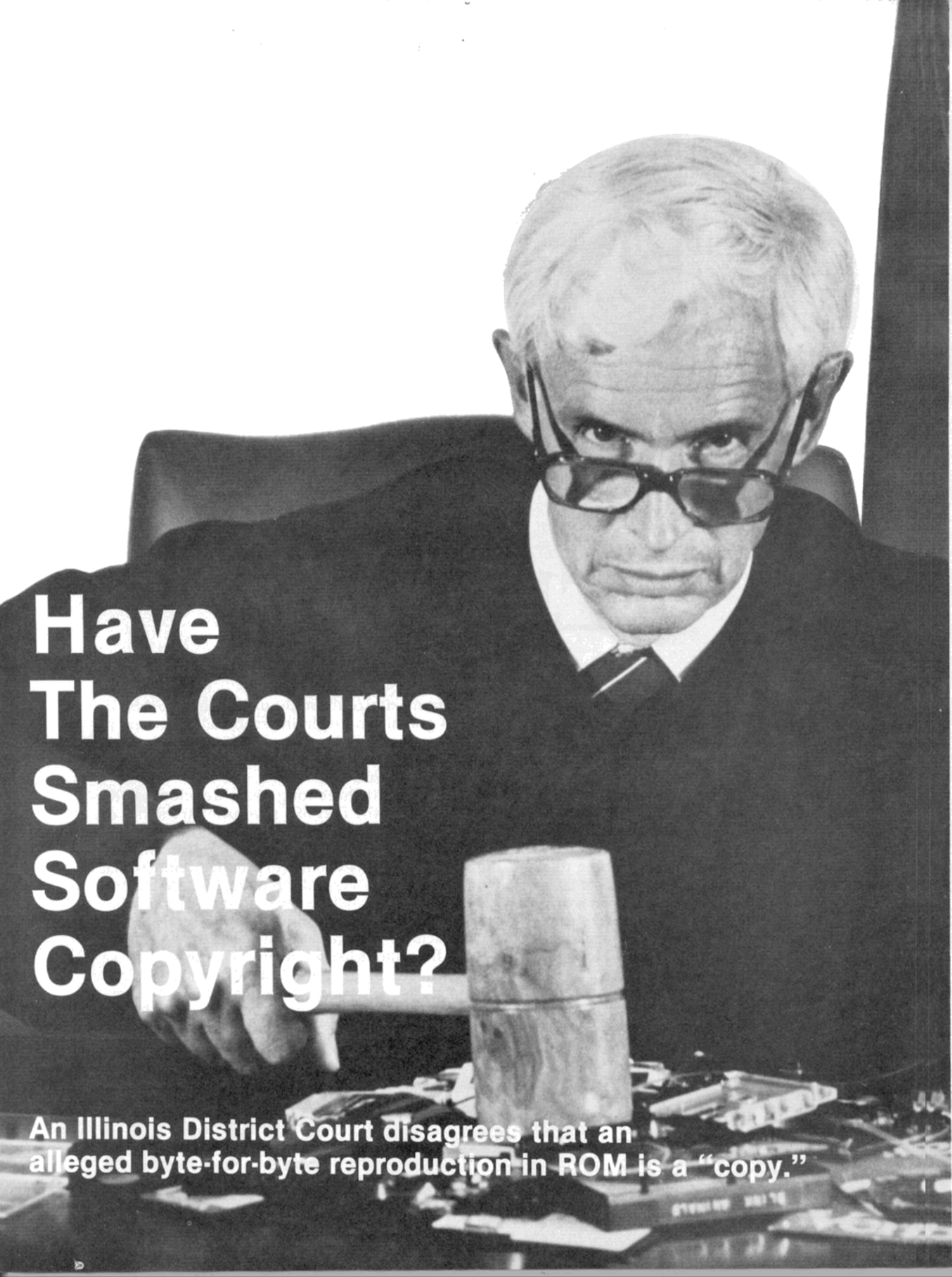
Reader Service ↙ 168

DIP Dot-matrix Printer

Pigskin's Graphic Display

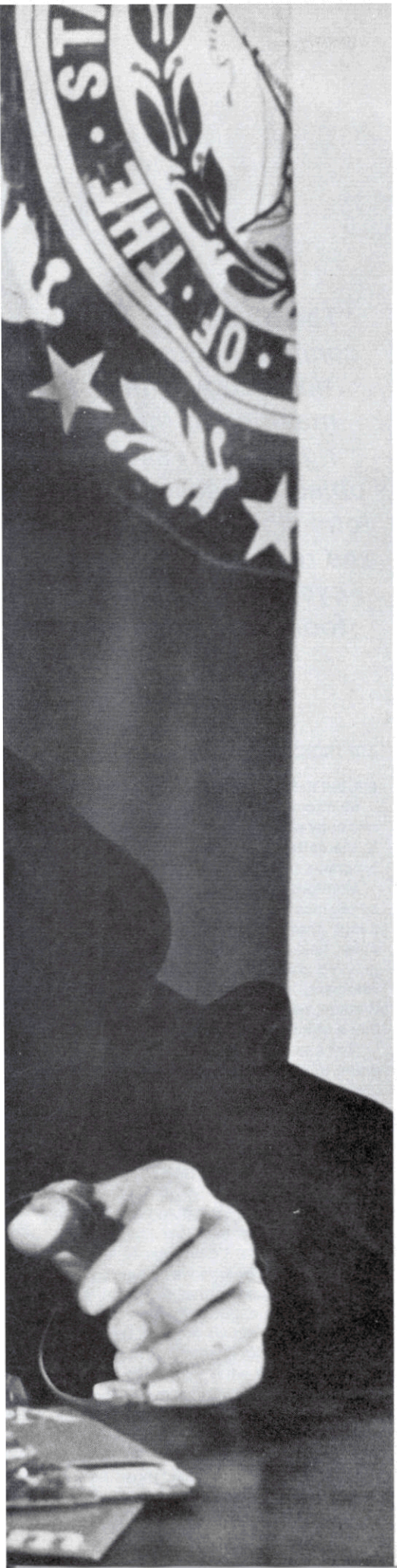


✓Reader Service—see page 226



Have The Courts Smashed Software Copyright?

An Illinois District Court disagrees that an alleged byte-for-byte reproduction in ROM is a "copy."



Dennis Bathory Kitsz
Roxbury, VT 05669

Back in the days of huge, marvelous and mystical mainframes, computer programmers spent years creating massive utilities and procedures for their customers. You've seen the pictures. Tied-and-jacketed men huddled over computer printouts with the kind of serious demeanor normally reserved for beleagured generals. But at the end of their work, they found security: the comfort of a job well done, the respect shown by appreciative and intimidated clientele, and, of course, substantial financial reward—all in good time, to be sure—but inevitable.

The year is 1980, and a change in the public's awareness now threatens that security and professional calm. The microcomputer has fallen from mystique to appliance, merely another contender for customer dollars in a clamoring marketplace.

What has programmers, software companies and software distributors so worried is an epidemic of program copying and trading that is very difficult to discourage under the current law. Stated very simply, the copyright laws, the most general form of protection for authors and artists, may not apply to the final versions of programs as prepared and sold on magnetic media—disks, tapes, wafers—or in read-only-memories (ROMs).

Dennis Bathory Kitsz is a composer, programmer and columnist for 80 Microcomputing. *He is an active defender of contemporary arts. His music has been performed in the United States and Europe. As an artist and a programmer, Dennis has been directly involved with the technical, aesthetic and legal aspects of the copyright issue for over a decade.*

The Copyright Law

The copyright law has a strong and impressive history of protection for authors and their works dating back to the eighteenth century, and rooted in common law. The Copyright Act of 1909 was framed around laws intended to protect published works, which quite plainly meant the printed word. The need to protect visual plastic arts (painting, sculpture) or technical arts (film, photography, sound recording) had hardly arisen. Oddities such as player piano rolls began to force the law to consider material that was not readily translated by the naked eye, but until recently the law depended mostly on judicial precedent.

The original copyright law was enacted in 1790, close on the heels of the Constitution itself, in order to protect "any map, chart book or books now printed." Step by step its coverage was extended to include designs, engravings and etchings, printed music, drama, photographs and negatives, sculpture ("statuary and models"), writings of an author, and motion pictures.

The 1909 copyright law came under fire in the 1960s, when piracy of records and tapes in the music industry became a serious issue. The law protected published sheet music, but how many people were flocking to stores to buy sheet music? The recording, the million-selling gold record, was king.

Executives of the recording industry called for copyright protection. But since the music performed by their artists was created during the performance or in the recording session itself, it wasn't written music. For the copyright office to require it to be translated into sheet music seemed inconsistent with the changes in the art form itself. Besides the composers, performers, arrangers and distributors were getting ripped off. Was the creative sweat of the music industry's artists any less valid, they asked, than that of typewriter-plunkers?

After more than sixty years, changes to the 1909 law were made to reflect changes in creative software. "Phonograms" became admissible for copyright, and all man-

ner of artistic, sculptural and film materials were legitimized.

The piano roll still didn't make it... nor did the computer program. The 1976 law did, however, introduce the term "works of authorship" as a term of general coverage.

At that time, the law provided that a commission on technological uses be set up to investigate two major problems with the act: The right of computer programs to copyright protection in their magnetic or ROM formats was unclear; and photocopying presented possible violations of copyright conventions.

The commission's final report called for copyright of computer programs. The report is nearly a year old, and, as this article is being written, the House Judiciary Committee has completed brief hearings on H.R. 6934, designed to implement some of those recommendations.

Within the industry an unspoken agreement for the most part has held back a flood of illicit copying, and authors and vendors have placed copyright notices on their works in the hope that some sort of grandfather clause might retroactively protect their works.

Then came *Data Cash* v. *JS&A.*.

The unspoken agreement fell to pieces in 1979, when Data Cash Systems, Inc. brought suit against JS&A Group, Inc., in Illinois Federal District Court. Data Cash claimed that JS&A took its Compu-Chess program and marketed it, byte-for-byte, as their own JS&A Chess Computer. The federal judge in the case held that the copyright law did not apply to the alleged ROM duplication, and denied Data Cash its motion for judgement against JS&A.

The question of laws governing unfair competition was deemed to be another matter, and presiding Judge Joel M. Flaum issued no summary opinion on that aspect of the case.

The judge's ruling was straightforward: "The parties have assumed that the ROM is a 'copy' of the computer program within the meaning of both the common law and the 1909 Act. The court does not agree. ...A 'copy' must be in a form which others can see and read."

Bill Gates didn't like the ruling. He is one of the creators of Level II BASIC, and the president of Microsoft, generally acknowledged the industry's software leader.

"It has a lot of people very upset. But as far as I'm aware, there's nothing (in the law) that eliminates that (copyright protection). Every manufacturer puts copyright notices on; Digital Equipment puts copyright notices on, we put copyright notices on...."

According to Gates, copyright laws are necessary to protect software from computer clubs or wherever people are likely to exchange it.

Said Gates, "If the law wasn't going to protect it, there wouldn't be any software written."

Just how extensive is the rip-off of programs in the home computer marketplace?

Large users' groups on both coasts maintain extensive libraries of programs for the purpose of sharing a cost burden; some such groups, but not all, discourage users from making their own copies from the library. New commercial libraries are beginning to advertise, with a "use fee" and an almost tongue-in-cheek "discouragement" of copying. Since a program is cheaper to copy than to buy, unlike a book, the program library becomes a tough problem for authors and vendors.

The End User

But the user/trader presents a unique threat to program writers. Software is expensive to produce and its market is severely limited by the number of home computers in use. How many programs does an author sell? Is it a "goldmine"? And how many sales are lost through gratuitous copying?

Bryan Mumford of Mumford Micro Systems doesn't care anymore. "I just decided it didn't matter."

"I sometimes wonder how much money I would make if I got paid for every copy of my programs that is being used. A lot more than I am now, you can be sure," says Mumford. "I do what I do, because I enjoy it, and if I start to get uptight about something like this, it stops being fun. Everyone isn't in this position, though. For most people, software sales are a strict and serious business. And bootleg copies are a big threat. Most computer people I know are upright moral people, but they can be bought pretty cheaply."

"The object phase of a computer program was not a copy within the meaning of the Copyright Act... since the object phase is not in a form which one can see and read with the naked eye, but a mechanical tool or machine part."

Judge Joel M. Flaum
Illinois District Court

Mumford says that he sells just about one copy of a program per town, and, shortly thereafter, that town has a plethora of programs from Mumford Micro Systems.

Mumford's organization is small and personal, much like the bulk of the cottage industry that has supplied the home computer field. But Intersystems is a large, growing company with lots of capital investment. Its president, James H. (Hank) Watson, believes clubs are responsible for much of his dollar losses.

"Let's face it," he says, "You sell (a program) to a quasi-computer club which has 250 members. To John Jones. And Bill Smith happens to buy Microsoft BASIC. And somebody else buys FORTRAN. I bet you dollars to donuts that everybody in the club has them within a week. What are you going to do?"

Hank continues, "It's going to be much more of an opportunity lost than a direct loss on my books; I could easily justify a million dollars. I'm sure we've only sold 50 percent of the copies that are in the field. That's a direct opportunity loss to me of $100,000 in the first six months. It's in that range, and it's a lot of goddamn money. And when you figure what your software time costs by the time you come up with the final product, that's five man-years. It comes to a cost of middle five figures! It takes a while to recover."

But for the average user, those figures seem vague. Neither Watson nor Mumford

can pinpoint specific losses, cash represented by actual copies they have seen in the hands of others. Are their fears perhaps imagined?

An ad appearing in the now-defunct *ONLINE* reads, "TRS-80: Swap quality disk software w/ doc? Send have/want lists to. . . ." Or how about this one: "TRS-80 program lending library. SASE. Exchange; Dept LL. . . ." Another claims, "TRS-80 'goodies.' Unique mix."

There seems to be an innocence about it, but what do these lists look like? One shocker comes in the form of (what else?) a computer printout, and contains more than 160 entries including RSM2D, REMODEL/PROLOAD, NEWDOS+, G2 LEVEL III BASIC, PIMS, FORTRAN, Misosys Disassembler, DESPOOL, TRCopy, Electric Pencil, Electric Secretary, General Ledger, Mailroom Plus—58 programs in this expensive commercial category alone—plus 109 games! This collection represents the majority of the finest software available to the personal computerist, developed over the course of years and totalling several thousand dollars in retail sales. All of it is exchangeable for items on the same trader's "want list." The wanted items also include some of the best: SARGON, COBOL, SCRIPSIT ("priority request!!!!" the trader notes), Infinite BASIC, System Doctor, Taranto's inventory programs, Electric Paintbrush, and 26 more.

The Quality Software Trader

With whom does this trader correspond? Where was he able to obtain dozens of programs? Who will fill his requests? The above-mentioned "quality disk software" trader, headquartered in the Northeast, contends that he and other traders are not only a positive force, but vital to the growth of the industry. He challenges the very premise of copyright protection.

"I reject, totally, the moral high horse that so many software vendors climb on. . . people are being *ripped off* in extraordinary amounts. *No one* cares about this or writes about it because the effect is so distributed and the media is controlled by the vendors, or at least by people who have a vested interest in software sales."

"It is oh-so-easy to preach about how bad it is to trade software," he continues. "The traders are an easy target, for sure. It's easy for an author to point to them as the 'reason he left the market.' And people will believe that. They (authors) don't suggest that their poor sales might be due to poor or overpriced software. No, the tacit assumption is that all software is worth its assigned price. Once that's accepted you can 'prove' almost anything."

The gentleman does concede that there are times when software trading is destructive: "When the person receiving the copy would certainly have bought it anyway, *and* would have gotten his money's worth, *and* doesn't take the money he saved and put it in another package. In all other cases (the *vast* majority), software trading. . . makes the participants better equipped and more likely to enjoy their machine and be productive authors themselves; draws someone into the hobby because of the software he can get; creates a sale that would not have occurred had the participants not been able to share the cost."

"If software trading was for some reason technically impossible. . . there would not be one-fifth as many authors."

A program trader

He continues, "Why do you think that 300,000 or so of these machines have been sold? Who do you really think buys all the software that is sold? If software trading was for some reason technically impossible, this system would not be one-fifth as popular as it is. There would not be one-fifth as many authors. . . . The fact is that software trading serves a real purpose in a hobby that is distribution-cost bound."

How will authors and vendors deal with this, indeed. Beyond the disagreement over the validity of program pirating, there are legal and philosophical entanglements as well. Over the years, industries as a whole have attempted to protect their developments in a number of ways:

- Patents. The embodiment of an original application or process is protected by this legislation, although obtaining a patent is time-consuming and costly. Computer programs have been generally excluded from this area of protection.
- Trade Secrets. A process or device whose workings are not released to the public is considered a trade secret. Any party obtaining the information is bound by the provisions of these laws. A non-disclosure agreement secures this secrecy for computer program vendors.
- Unfair Competition. Making a profit from the work of others is viewed as unfair competition. State laws vary widely on this issue; program authors with the economic wherewithal can pursue this route.
- Copyright. The expression of ideas in a human language is protected by this law. Registration with the copyright office secures universal copyright as well, which provides international protection in most non-Soviet countries.

It has become all too obvious to members of the "cottage" software industry that two of those protections, trade secrecy and unfair competition, provide little help. They are simply too costly for the small corporation, partnership, or individual author to employ to safeguard computerized works.

A large company like Microsoft, on the other hand, depends heavily on the idea of unfair competition. Bill Gates says, "We spend millions of dollars a year creating software programs, and we are protecting those in several ways. There's the trade secret laws where we get non-disclosure—

that's how we handle our source codes and our so-called commercial packages that are high-priced. But for our low-cost software, we simply can't do that.... But, if a trade secret is released, and people are taking advantage of (it), they are subject to a penalty."

"Our code is the trade secret," says Gates. "We're not giving it away, we're selling it, just like Coca-Cola; they license it. Or Dow Chemical...making ethylene. It's a trade secret, so those that have the process pay. They sign the non-disclosure, just like the people who receive our software sign our non-disclosure."

Gates differentiates between questions of fact and law. Legal remedies are only useful after it has been proven that the program has been stolen.

"You're talking about a thing that is *sixteen thousand* instructions with 256 possibilities for each one. I can certainly prove, *if* it's derived from my work, *that* it's derived from my work!... In the case of software, they (the court) would rely on expert testimony. If somebody's camouflaged the thing pretty well, that's a question of fact, not of law. The question is, did they borrow from my work. Okay, assume that I can prove that and convince the court that they borrowed from my work, then you have the question of law, what am I going to do about that?"

Intersystems' Hank Watson is looking for a solution. "If you even do find somebody, you publicly crucify them and hope that that is some deterrent. That's the only tack you can take, but no one has been successful in general. What we'd like to do is protect the concept."

But is the copyright law applicable to computer programs? Is it enforceable? Moreover, is it a desirable means of protecting program authors? Judge Flaum's decision against Data Cash Systems opens a wide gap between the law's explicit coverage and the recommendations of the National Commission on New Technological Uses of Copyrighted Works (CONTU), set up in 1975 and whose work was completed in July, 1978.

In question is Section 117 of the 1976 law, which states in part, "...this Title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information,...than those afforded to works under the law...."

"The purpose of Section 117," in the words of Judge Flaum, "is to preserve the status quo. It is not intended to cut off any rights that existed on December 31, 1977, or to create new rights that might be denied under the predecessor to the 1976 Act, the Copyright Act of 1909...or under common law principles."

One part of Judge Flaum's opinion in *Data Cash vs. JS&A* echoed the dissent of CONTU Commissioner John Hersey, novelist and chairman of the Author's League of America. Said Hersey, "Every program comes to fruition in its mechanical phase."

Judge Flaum in his decision nearly a year later, concurred, "Normally, a computer program consists of several phases," writes Judge Flaum, in an opinion which derives from Hersey's dissenting one. "The first phase is the development of a flow chart which is a schematic representation of the program's logic. The second phase is the development of a 'source program,' which is the translation of the flow chart into computer programming language.... The third phase is...an 'assembly program' which is a translation of the programming language into machine language, i.e., mechanically readable language."

The crux of the judge's decision is summarized in his next few sentences: "...Assembly programs are virtually unintelligible except by the computer itself. Finally, the fourth phase is the...'object program' which is a conversion of the machine language into a device commanding a series of electrical impulses. Object programs... cannot be read without the aid of special equipment and cannot be understood by even the most highly trained programmers."

With that in mind, Judge Flaum concludes, "Thus, at some point in its development, a computer program is embodied in material form and becomes a mechanical device which is engaged in the computer to be an essential part of the mechanical process."

CONTU, on the other hand, collectively felt that computer programs should be afforded the protection of the law, but conversely should not overly burden end-users. Thus, the Commission stated that copyright should forbid unauthorized copying of computer programs, should not inhibit their rightful use, should not prevent development or distribution, but should, in its words, "not grant anyone more economic

Reader Service—see page 226

power than is necessary to achieve the incentive to create." It proposes to eliminate the present ambiguous Section 117, and replace it with a new section specifically detailing those protections and limitations.

H.R. 6934, the bill now pending in the House of Representatives, will in part do that, defining a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." The bill drops the old Section 117, and includes new wording authorizing the user to produce copies only if "such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful."

Nevertheless, H.R. 6934 does not define what constitutes a "copy." CONTU recommended protections that should clearly include magnetic tape, disks, and other related forms, as well as ROM chips themselves. The proposed law still leaves that decision to the courts.

Failure of Secrecy

One of the incentives that even CONTU found it had to address was the question of the small-scale programmer. It pointed out failings in the trade secrecy laws: "Because secrecy is paramount, it is inappropriate for protecting works that contain the secret and are designed to be widely distributed. . . . It substantially precludes the use of trade secrecy with respect to programs sold in multiple copies over the counter. . . ."

Finally, CONTU reported that unfair competition laws were neither nationally uniform nor applicable to situations in which unauthorized copying was being done without a profit motive—certainly the case in the "share it with a friend" cycle.

The Commission, however, was not unanimous. Its most comprehensive critic was John Hersey, one of CONTU's three dissenters on the matter of computer copyright. Hersey argues that a computer program in its final form is much more like the cam in a machine than the work of an author. His argument is straight to the point: "Printed instructions explain *how* to do something; programs are *able* to do it."

Hersey dismisses any analogy between computer programs and recorded music as well, claiming that true works of authorship may be "fixed" in many forms, yet their main purpose is communication among human beings. "But a program," he emphasizes, "does not communicate information of its own, intelligible to a human being. It utters work. Work is its only utterance and its only purpose. . . The mature program is purely and simply a mechanical substitute for human labor." Not a very pleasant thought for the many programmers who consider their works to be gems of creativity, efficiency and inspiration.

Commissioner Hersey's dissent urges separate legislative protection for computer programs, but insists that they are "the embodiment of a system or process," and not a description of it, and from that point of view are legally ineligible in light of the underlying principles of copyright law.

The majority of programs are written in so-called "high-level" languages which are not compiled, where the object code so crucial to the arguments of John Hersey and Joel Flaum never appears.

A program on the TRS-80 home computer is created in that ugly but serviceably "human" language, BASIC. Admittedly, there is some translation and condensation that goes on while the program resides in the microcomputer, but when viewed intact on the video screen, the program looks like, reads like, and is constructed like the program created by the program, not a "mechanical part."

Things have been further clouded by the introduction and development of digital recordings and computer music. What part of the music or recording is "data base" (copyrightable) and what part is "program" (not eligible for copyright)? The distinction between data bases and programs is nebulous as well; every computer operates from a data base, as most computer instructions have operands and no useful program can function without them. Whether these operands are already embedded in the program or are accessed externally, they remain integral parts of any computer action.

The processor, the microcomputer's controlling chip, does not know the difference, and cannot know the difference. Only a human can define the distinction between data base and program by reading the code or examining some manifestation of the bits and bytes. Every machine language programmer knows the feeling of having a program crash because it misguided the computer.

Perhaps it is the human distinction alone that is the crucial one, not the machine distinction.

MULTI-USER OASIS HAS THE FEATURES PROS DEMAND. READ WHY.

(THEN COMPARE.)

Computer experts (the pros) usually have big computer experience. That's why when they shop system software for Z80 micros, they look for the big system features they're used to. And that's why they like Multi-User OASIS. You will too.

DATA INTEGRITY: FILE & AUTOMATIC RECORD LOCKING

The biggest challenge for any multi-user system is co-ordinating requests from several users to change the same record at the same time.

Without proper co-ordination, the confusion and problems of inaccurate or even destroyed data can be staggering.

Our File and Automatic Record Locking features solve these problems.

For example: normally all users can view a particular record at the same time. But, if that record is being updated by one user, automatic record locking will deny all other users access to the record until the up-date is completed. So records are always accurate, up-to-date and integrity is assured.

Pros demand file & automatic record locking. OASIS has it.

SYSTEM SECURITY: LOGON, PASSWORD & USER ACCOUNTING

Controlling who gets on your system and what they do once they're on it is the essence of system security.

Without this control, unauthorized users could access your programs and data and do what they like. A frightening prospect isn't it?

And multi-users can multiply the problem.

But with the Logon, Password and Privilege Level features of Multi-User OASIS, a system manager can specify for each user which programs and files may be accessed—and for what purpose.

Security is further enhanced by User Accounting—a feature that lets you keep a history of which user has been logged on, when and for how long.

Pros insist on these security features. OASIS has them.

EFFICIENCY: RE-ENTRANT BASIC

A multi-user system is often not even practical on computers limited to 64K memory.

OASIS Re-entrant BASIC makes it practical.

How?

Because all users use a single run-time BASIC module, to execute their compiled programs, less memory is needed. Even if you have more than 64K, your pay-off is cost saving and more efficient use of all the memory you have available—because it services more users.

Sound like a pro feature? It is. And OASIS has it.

AND LOTS MORE...

Multi-User OASIS supports as many as 16 terminals and can run in as little as 56K memory. Or, with bank switching, as much as 784K.

OASIS IS AVAILABLE FOR

SYSTEMS: Altos; Compucorp; Cromemco; Delta Products; Digital Group; Digital Microsystems; Dynabyte; Godbout; IBC; Index; Intersystems; North Star; Onyx; SD Systems; TRS 80 Mod II; Vector Graphic; Vorimex.

CONTROLLERS: Bell Controls; Cameo; Corvus; Konan; Micromation; Micropolis; Tarbell; Teletek; Thinkertoys; X Comp.

Multi-Tasking lets each user run more than one job at the same time.

And there's our BASIC—a compiler, interpreter and debugger all in one. An OASIS exclusive.

Still more: Editor; Hard & Floppy Disk Support; Keyed (ISAM), Direct & Sequential Files; Mail-Box; Scheduler; Spooler; all from OASIS.

Our documentation is recognized as some of the best, most extensive, in the industry. And, of course, there's plenty of application software.

Put it all together and it's easy to see why the real pros like OASIS. Join them. Send your order today.

MAKES MICROS RUN LIKE MINIS.

OASIS

CIRCLE WHAT YOU WANT

Product	Price with Manual	Manual Only
OPERATING SYSTEM (Includes: EXEC Language; File Management; User Accounting; Device Drivers; Print Spooler; General Text Editor; etc.)		
SINGLE-USER	$150	$17.50
MULTI-USER	350	17.50
BASIC COMPILER/ INTERPRETER/DEBUGGER	100	15.00
RE-ENTRANT BASIC COMPILER/INTERPRETER/ DEBUGGER	150	15.00
DEVELOPMENT PACKAGE (Macro Assembler; Linkage Editor; Debugger)	150	25.00
TEXT EDITOR & SCRIPT PROCESSOR	150	15.00
DIAGNOSTIC & CONVERSION UTILITIES (Memory Test; Assembly Language; Converters; File Recovery; Disk Test; File Copy from other OS; etc.)	100	15.00
COMMUNICATIONS PACKAGE (Terminal Emulator; File Send & Receive)	100	15.00
PACKAGE PRICE (All of Above)		
SINGLE-USER	500	60.00
MULTI-USER	850	60.00
FILE SORT	100	15.00
COBOL-ANSI '74	750	35.00

Order OASIS from:

Phase One Systems, Inc.
7700 Edgewater Drive, Suite 830
Oakland, CA 94621
Telephone (415) 562-8085
TWX 910-366-7139

NAME _____
STREET (NO BOX #) _____
CITY _____
STATE _____ ZIP _____

AMOUNT $ _____
(Attach system description; add $3 for shipping; California residents add sales tax)
☐ Check enclosed ☐ VISA
☐ UPS C.O.D. ☐ Mastercharge
Card Number _____
Expiration Date _____
Signature _____

What, then are the options? The law, of course, could be more defined to include software, more specifically than H.R. 6934, but the legal problems would linger for years as courts find and set precedents.

Another proposition is to provide convenient licensing companies that would handle royalties for program authors. It is a method which has worked with great success in the music industry, where such giants as the American Society of Composers, Authors and Publishers (ASCAP) and Broadcast Music, Inc. (BMI), follow the trail of recorded music through radio transmissions, in jukeboxes, and on retailers' shelves. How do program vendors view this possibility?

Hank Watson of Intersystems says, "That's a little bit different. They're talking about millions of copies of stuff . . . I think it's going to take a while to evolve. Going back to the fifties, there were lots and lots of pirates and there were fewer in the seventies. It's a process we'll all have to go through."

Watson's distinct lack of enthusiasm was echoed by others in the industry, who believed that copying would likely continue on the amateur level, and the volume of program sales would never be high enough to merit a network of licensing organizations.

Certainly, major publications in the field have been lax about discouraging software piracy. Almost alone in this crusade is Wayne Green, publisher of *80 Microcomputing, Kilobaud Microcomputing, 73 Magazine* and owner of Instant Software. Green's weapon is cash—ten thousand dollars of it, up front—to reward the first person to turn in a major ISI software pirate for successful prosecution.

Has anyone tried to take him up on it? Green replies, "No, but to some degree the intent is working—it is having an impact with dealers, but I don't know how much it's impacting clubs."

Other publications have for the most part held their editorial tongues, offering an occasional fingerwagging, but sticking to safer, professional subjects such as the impact of computer technology on cartography, or why Pascal is the next standard computer language.

Whether a forced infusion of morality to counter illicit duplication would have any effect is not discernible; those out to turn a profit by exploiting the work of others will hardly heed cries from editors, if the copyright law remains ineffective.

cont. to pg. 64

"Yes, surely, some computer programming is wonderfully creative. But copyright was not designed to protect the products of creativity as such; it was designed to protect literary works . . ."

John Hersey
Author and CONTU Commissioner

In recent correspondence, John Hersey expanded his dissent. "Yes, surely, some computer programming is wonderfully creative. But copyright was not designed to protect the products of creativity as such; it was designed to protect 'literary works.' The designer of the cam was immensely creative, too; the inventor of the wheel was a genius—would that we knew his name to thank him! But we don't copyright those things."

The question of musical performances, especially with the advent of electronic music, was again posed to him in light of a program being analogous to a cam.

The musical "instruction set" consists of representations of specific actions to be taken, very much like those of a cam, and the untrained individual cannot "read" these instructions into the mind's ear. A very, very few highly trained specialists can.

The fingers must push "up-down" and the tongue and lungs go "in-out," or there is silence from the winds; the arms must go "back-forth" while the fingers go "up-down" or the strings are quiet; and so forth.

After much harangue about objectivity and a sizable long distance bill, the editors have succumbed to Kitsz' right to a philosophical digression.

Very little music is strictly theoretical; it is a set of printed instructions to produce the final sound, no matter whether that reading is done by a violinist or by an optoelectronic score reading device. The score itself is not the music, but a very digital-looking analog of the music, directly accessible only a few highly skilled "readers."

Hersey disagrees, believing "a musical score is *not* analogous to a program. The score tells the human performer what fingers to push 'up-down' on wind instruments, and when the lungs should go 'in-out,' and what the fingers should do on the strings. In the case of the computer program, the instructions become part of the machinery and make the 'up-down' and 'in-out' take place."

As for the composer of computer music, Hersey again takes the opportunity to distinguish between the human and the machine. "I would certainly think that the electronic composer's music, or score if that is the product—whatever *issues from* the computer and can be perceived by the human ear can—should be copyrightable. It manifestly is. *It* is the product of the composer's creativity. He may also have been creative in manipulating the machine to produce the music, or the printout, but it is my view that that sort of creativity can be and should be protected under other laws than copyright."

Melville B. Nimmer teaches copyright and constitutional law at UCLA, and was Vice-Chairman of CONTU. While sharing some of Hersey's reservations, he is prepared to distinguish between two general types of computer programs. The first would be eligible for copyright, and would include "works which themselves qualify for copyright protection . . . . On the other hand, programs which control the heating and air-conditioning in a building, or which determine the flow of fuel in an engine, or which control traffic signals, would not be eligible for copyright because their operations do not result in copyrightable works."

He then departs considerably from Hersey's point, claiming that the distinction is "consistent with the recognized copyrightability of sound recordings. It sometimes has been argued that while printed instructions tell *how* to do work, computer programs actually *do* the work. But this is also true of sound recordings, which in a

sense constitute a machine (the phonorecord) communicating with another machine (the record player)."

As a whole, CONTU defended the concept of programs as works of authorship. In fact, the defense included some remarkably colorful, if dubious, considerations about those items presently eligible for protection.

"Traditional works have led to processes both more rigid and more flexible than those to which computer programs lead. When a phonorecord or motion picture is used in conjunction with a properly working machine, the same result will occur on the first, the second, or the thousandth running. The chorus will remain silent until the fourth movement of Beethoven's Ninth Symphony, and Bogart will stay in Casablanca forever.... The process is virtually immutable. That is less true when a program is used, since it contains alternative branches selected only after use has begun..."

Artists and philosophers would have a hard time with those claims, however attractive they may be in the defense of a program author's right to legal protection. Subtle or gross differences of acoustics, speed and fidelity dramatically alter that "immutable" recording of Beethoven—the recordings can be played in monaural, stereo, or synthetic quadraphonic, or on small players with styli suitable only for sewing. Bogart's trysts may be obscured by faded film, a geriatric projection bulb, variations in shutter speed and screen size.

By contrast, a computer program is only reliable, only viable, in a trustworthy medium with an accurate CPU and stable memory. Its speed may be transformed, perhaps by the system's clock, but the forgiving flexibility of even the "fixed" art forms is unwelcome in the precise world of computer activity.

Philip K. Hooper is a programmer, author and authority on 6502-based systems. Hooper notes that at some point the interaction is initiated on the human level. "If one is referring to the program and its preset data in the computer, then there is no case (for copyright), just as one assumes that the bands on a record are in a certain order. Only the breakpoints introduced by user interaction really put the computer into a different realm. Human interaction produces a greater variety of *consequence streams*."

The philosophical questions are not peripheral to the copyright issue; they are in fact at the heart of it. All parties seem to be agreed when the query is posed, "should computer programs be afforded protection under the law?" The legal and humanistic storms are born by asking, "which laws shall protect them?"

Can the copyright law protect the computer program? Bill Gates believes he can recognize his program, "camouflaged" or not. But can copyright law distinguish the difference?

There are many methods of camouflage; some can well be defined as translation into a foreign language, while others involve encoding in the manner of a secret message. Translation into a foreign language is not a difficult problem, and traditional works have long been protected in that realm. But take for example, the following 16 bytes of Level II BASIC machine code. These are the opening instructions copyrighted by Microsoft:

```
F3 AF C3 74 06 C3 00 40 C3 00 40 E1 E9 C3 9F 06
```

In the form presented here, and by means of any Z-80 disassembler available, these instructions can be transformed into the mnemonics of:

```
DI
XOR   A
JP    0674
JP    4000
JP    4000
POP   HL
JP    (HL)
JP    069F
```

We have here visible, human-readable code. It is unremarkable and undistinguished as a small excerpt, but a large block would be clearly identifiable as Level II BASIC. Below are another 16 bytes of hex code:

```
79 D7 E1 BA 03 60 80 20 61 80 20 70 E4 E1 CF 83
```

This code translates into the following interesting mnemonic instructions:

```
LD    A,C
RST   10
POP   HL
CP    D
INC   BC
LD    H,B
ADD   A,B
JR    NZ,$+63
A,B   JR NZ,$+72
CALL  PO,CFE1
ADD   A,E
```

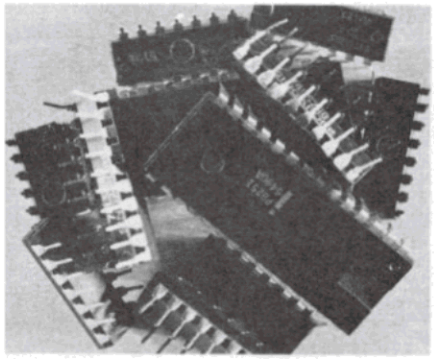
This is useful code as well, and a careful first examination of this code might reveal no relationship whatever between the Level II excerpt and the second block of bytes. However, one need only present the first four bytes of each and convert them to binary to discover the ploy:

```
11110011101011111100001101110100
01111001110101111110000110111010
```

Each bit has been rotated to the right one position! It would take not more than a dozen bytes at the entry point of the "camouflaged" program to rotate an entire 12K copy of Microsoft's BASIC one bit back to the left, and there it would be, ready to use. The magnetic copy, moreover, would not match the original "stolen" program in format, parity, checksum, nor byte pattern.

It does not become a copy until it is under the user's control! Who, then, is the offending party? No translation was effected; rather, in its magnetic, object and disassembled versions, it is a completely new program.

Oure yesw ouldn oth avet oom ucht roubled iscerningt hatt herei ss omes orto fe ncodingb eingi mplementedi nt hiss entence. In fact, concerning literary works, this manner of poetic license is normally allowed under the concept of artistic freedom of manipulation. The structure of the sentence has been so changed that a normal reading is rendered virtually impossible. An identical process was employed in rotating those 16 bytes, and direct, unmodified execution of the code would in no way lead in the same direction as does Level II BASIC.■



Microsoft's Bill Gates feels that the wording of the law should merely "make it clear that the copyright law covered [magnetic media]. It is *not* an extension. Look at the thing that commissioned CONTU to start with," says Gates. "It doesn't say that the law doesn't cover these things; it says that they've been asked to come up with a clear position clarifying the exact procedure."

House Judiciary Committee counsel Bruce Lehman reports that H.R. 6934 is likely to become law during this session of Congress. The bite of this law, if any, may not be felt for some time, based upon the bill's generalized wording. The entire document, the "Computer Software Copyright Act of 1980," is a single slip of paper, and the legal entanglements are hardly unsnarled by its ambiguities.

The Lock and Key Method

So the problem waits for a legal solution while programmers and vendors continue to seek refuge in the lock-and-key method. Major programs are provided with special loaders which must gain control of the microcomputer before the remainder of the program may be fed into the machine. The first and best known of these loaders is provided with Microchess, introduced for the TRS-80 two years ago. The loader is almost, but not quite, identical to the normal TRS-80 SYSTEM command; it was therefore the easiest lock to open even for amateur machine language enthusiasts.

More sophisticated loaders began to appear, including those from such vendors as The Bottom Shelf and ABS Suppliers. These loaders actually alter the baud rate at which data is input to the computer. Disk programs, already protected to some extent by the difficulty of working directly with the disk's data transfer system, lend themselves even more so to security—the loader effectively disappears once the program is in place. At disk speed, this process is virtually instantaneous and quite opaque.

Others, like Electric Pencil, block-move themselves all around memory in a game of electronic hopscotch. But, as one reader comments, "Aside from legal or ethical questions, how are you going to stop pirating of software? . . . If by some means a program is made 'uncopyable,' someone will figure out a way to copy it sooner or later."

Even Wayne Green agrees: "We prefer to make everything as simple as possible—but on the other hand, there are other programs that can decipher anything you can do."

Bill Gates speaks defensively and excitedly of his own company's reactions: "Looking at the amount of software we offer, we are the most ripped-off company around, because we offer a broad range, and we try to offer it for these low-cost computers. And we view the thing totally as an experiment. If there aren't enough honest people out there to buy the stuff, we'll end it. Most of our packages we won't put down at the low end."

In its advertising, Microsoft bills itself as setting the industry standard. Gates claims that his "experiment" can be ended at any time. But CONTU addresses the issue of corporate size, concluding that the social effects of the current copyright law's ambiguities are more acceptable to the individual rather than the corporate author/programmer.

CONTU even questions whether the independent will really benefit from an extension of copyright protection, asking rhetorically, "why do the large industrial corporations press for copyright?" The Commission answers itself with caution, suggesting that copyright protection might tend to reinforce the dominance of the large corporations over the small, independent houses. This was borne out by the influences of lawyers and patent specialists representing

"Let's face it. You sell one (a program) to a quasi-computer club which has 250 members. To John Jones . . . I bet you dollars to donuts that everybody has it within a week."

Hank Watson
Intersystems

large computing firms during the CONTU hearings.

Here, the subtle debates over "works of authorship" fall victim to the harsher activities of supply and demand. There comes the realization, finally, that Data Cash sued JS&A not to pursue a philosophical or moral crusade, but to prevent JS&A from earning a profit on Data Cash's work.

The house of cards has begun to fall as a result of the JS&A decision. One other case that may be affected involves Nestar Systems. Nestar Systems, based in Palo Alto, has asked for a restraining order against a European company which plans to market for the PET a package called The BASIC Programmer's Toolkit. Nestar alleges that the unnamed Netherlands-based organization plans the marketing of cassette or disk versions of its ROM product, and the JS&A decision has Nestar concerned about its rights.

Harry Saal, president of Nestar, is unclear about Judge Flaum's decision in the JS&A verdict, for Saal claims that "a human was very capable of understanding the 1's and 0's." The differentiation Flaum had made, of course, was that the eligibility of machine-readable code was in question, not its translation into ones and zeros.

"It's simply our feeling, and I believe it will be shared by everyone else who reads *80 Microcomputing*," says Saal, "that the act of creating software is a development process that needs to be protected somehow. There must be a means by which people, having performed their work, can be free to market their efforts without some-

one else coming along and copying their material and making a profit based on the simple cost of reproduction.

"We're looking here at a product; a serious amount of analysis and thought and market planning and serious investment of money goes into the development of it, and every time we look at a product in the future, we have to go through that same procedure. It's going to be very difficult (in the future) to justify software development."

In conclusion, it is the voice of Bill Gates that rings in the ear. He was asked to react to a recently published disassembler handbook from Richcraft Engineering.

Interviewer: The author provides virtually all of your code in hex, out of which he's taken a few bytes. That's to make sure that the person bought the original product.

Gates: He's got our code in hex?

Interviewer: Your entire code less maybe two dozen bytes he took out specifically to say he didn't violate your rights by providing the whole code.

Gates: He certainly violated our rights!

Interviewer: He feels he has not . . .

Gates: He certainly has, because that's my material! Whose does he think it is? Does he think that he has the right to go out and commercially profit by republishing something that we created? I mean, that's ludicrous! Why should he be making money from that? All he did was take our stuff! ■

A Selection from the "Connecticut Trader's" Blackmarket List

Software Available:

Utilities

1. RSM2D (SSS).*
2. REMODEL/PROLOAD (RACET).* 3. GSF/DOSORT (RACET).*
4. MEMORY TEST. 5. NEWDOS+.*
6. COPY (Copy System Tapes, from RACET).*
7. DCU (SSS).* 10. G2LEVIII.* (Microsoft)
11. BUGGER (Radio Shack DISK DEBUG relocated to operate at upper memory - Specify memory configuration.)
12. KVP (Micklus).* 13. PIMS (SCELBI), Disk & Tape.
14. UPRLWR (Upper/lower case software; supports upper/lower case in BASIC. Uses Electric Pencil hardware. Tune shift for upper case; correct printer output. Self prompting. Fits all memory sizes.)
15. IDM III (File management system, Micro-Architect).*
16. CPM & CBASIC. (Buy Documentation from Lifeboat). (2 Disks. My personal opinion: CPM is a bummer.)
17. FORTRAN. (Buy Documentation from Lifeboat). (2 Disks).
18. ST80D (Micklus).* 19. TOOL KIT.* (TBS)
20. DOS 3.0 (Unprotected).*
21. MAILIII (Micro Architect).*
22. MISOSYS DISASSEMBLER (Soltoff).*
23. PCLEND (ALS, works with DOS 2.2 and NEWDOS).*
24. TELCOM (HDS).* 25. TRASH.(ACS).*
26. TIMESER (RACET, Multiple Regression).*
27. PERCOM MICRODOS.
28. COMPROC (RACET).*
29. COMMAND (Similar to COMPROC).
30. DESPOOL (Runs printer & computer at same time).*
31. TRCOPY.
32. EITEST (EXPANSION INTERFACE TEST).
33. ATERM (Acorn).*
34. PILOT (Edison, Mad Hatter.)*
35. TRS232 "FORMATTER" (SSS, Advanced Printer Package).*
36. FORTH (Programma).*
37. FLEXL & TDISK (Acorn).*
38. FIX (SOFTWARE DRIVER FOR HEATH WH-14 TO TRS-80).
39.
40. ESP1 (SSS).*
41. REGRESSION.
42. DESCRIPT (STATISTICS).

Word Processing

1. ELECTRIC PENCIL (Disk & tape). Also have a version that works with NEWDOS.*
2. ELECTRIC SECRETARY (BASIC Word Processor - Works with Electric Pencil Upper/Lower Case hardaware mod.).*
3. WORDIII (Micro Architect).*
4. TEXTEDIT: A BASIC TEXT EDITOR (H. S. Howe).*

Business

1. MAILING LIST (Radio Shack).*
2. INVENTORY CONTROL SYSTEM (Radio Shack, ICS).*
3. GENERAL LEDGER (Radio Shack).*
4. OSBORNE: AR, AP, GL, PAYROLL. (Buy Osborne books for documentation.). (5 Disks).
5. BIZ80 NAME/ADDRESS LIST.* 6. BIZ80 INVENTORY.*
7. BIZ80 GL.*
8. MAILIII (Micro Architect).*
9. ADVANCED PERSONAL FINANCE (Micklus).*
10. CHECK REGISTER ACCOUNTING (Personal Computing).
11. INVENTORY (AJA) (Need Documentation).
12. PAYROLL (AJA) (Need Documentation).
13. AR (AJA) (Need Documentation).
14. AR (Hebbler, TSE).*
15. MAILROOM PLUS.*
16. CHECKBOO/BAS.

Games

1. TIMETREK.*
2. ZCHESS (Need Documentation).
3. BACK40 (Need Documentation).
4. MUSIC MASTER.*
5. POKER 6. BARTEND 7. TAROT
8. BRIDGE (DUISMAN).
9. STRIKEZONE (BASIC + Obj. file baseball).
10. FASTGAMMON.* 11. MUSIC (Radio Shack's).*
12. SANTA PARAVIA etc. 13. BALLOON CROSSING
14. KALEIDASCOPE 15. EIGHTOUT 16. ZARBOR
17. FOOTBALL 18. REVERSE 19. FROGS.
20. MAZE (DOG'S) 21. SKETCH 22. GRAPHIC
23. CONCENTRATION (MICKLUS).
24. ELIZA 25. CHIEF
26. UFO 27. ATOM20 28. ZODIAC
24. BOWLING (TSE) 30. JIGSAW 31. AROUND THE HORN
32. STARWARS 33. GOLDMINE 34. HORSE RACE
35. CRIBBAGE 36. SLOT
37. LIFE (CHRISTOPHERSON, with sound)
38. TICTAC 39. SIMON (sound) 40. TREKIII (Micklus).*
41. SPILOT 42. DARTS 43. PINBALL
44. NIM (with sound) 45. SNAKE (with sound)
46. OTHELLO 47. SLOT4 48. SNOOPY (Picture)
49. PILLBOX 50. ROBOT 51. AIRAID.
52. ADVENTURE 1,2,3,4,5 (disk). 53. MONOPOLY.
54$ INVASION (Radio Shack)*. 55. SAUCER (Radio Shack)

Programs Wanted:

1. Any CPM business.
2. RADEX10
3. COMPRESS IT (Bluebird)
4. SIMPLIFY IT (Bluebird)
5. Any Statistics programs
6. PROGRAM INDEX (MUMFORD MICRO SYSTEMS)
7. SARGON II.
8. SUPERMAIL (OKLAHOMA COMPUTER CO.)
9. TARANTO Inventory
10. X-WING FIGHTER II (TSE).
11. Anything put out by the CPU Shop (Kilobaud, Aug.79 p.132)
12. COBOL (FMG).
13. WARLORDS
14. MAGSAM III (Kilobaud, Aug.79 p. 143)
15. DR. CHIPS
16. RADIO SHACK'S NEW WORD PROCESSOR (PRIORITY REQUEST !!!!).
17. BASIC COMPILER (MICROSOFT).
18. SYSTEM DOCTOR (TBS).
19. SELECTOR IIIC2 (From MICRO-AP; also sold by LIFEBOAT)
20. KVP232 (MICKLUS)
21. Any BIZ80 Business programs (Aside from NAME/Address List, GL, & Inventory Control System).
22. DISCIO (Barden, Sold by CHICATRUG).
23. NEW NEWDOS (3.0)
24. INVASION ORION
25.
26. INFINITE BASIC (both parts) from RACET (PRIORITY REQUEST !!!).
27.
28. ADVANCED STATISTICS PACK (Radio Shack)
29. SPACE BATTLES
30. MONITOR 4 (ACS).
31. PIE (Programma).
32. BLOCKADE (Personal Software).
33. ELECTRIC PAINTBRUSH (Personal Software).

Programs with * come with documentation.

THE TIME SAVER...

JPC Products Announces a New High Speed Cassette System
"The Poor Man's Floppy" for the TRS-80*
$90 Kit - $120.00 fully assembled.

The TRS-80* is a great small computer. But its cassette recording system can be very frustrating, particularly if you can't read an important cassette. JPC Products Company has developed an improved cassette system that uses your present cassette recorder but operates much faster with better reliability. The TC-8 plugs into the expansion connector on the back of the keyboard and saves and loads 5 times faster! Less than ONE BAD LOAD in a MILLION BYTES! With the VOLUME CONTROL ANYWHERE BETWEEN 1 AND 8. The TC-8 is available in an easy to assemble kit or fully assembled. JPC has an exclusive "can't fail" kit guarantee. If you build the TC-8 and for any reason it doesn't work, we will make it work at NO COST. All you have to pay is the shipping. We guarantee it. The TC-8 magic is partly done in software. So you have to load a small program in upper memory. It is usually out of the way there. We provide the software on a cassette that comes with the TC-8. Just load it in. Here's how you order. Send $90.00 for the kit ($120.00 fully assembled) plus $3.50 postage and handling to JPC Products Co., 12021 Paisano Court, Albuquerque, NM 87112 (N.M. Res. add 4% sales tax.) Credit card orders accepted by phone or mail.

JPC PRODUCTS CO.

✓190

*Trademark of Tandy Corporation

Phone: (505) 294-4623 12021 Paisano Ct. Albuquerque, N.M. 87112

✓Reader Service—see page 226

Part Two of a Series

The essence of variables.

Into the 80's

Ian Sinclair
89 Alexandra Road
Sible Hedingham
Halstead, Essex CO9 3NP
England

It doesn't take long for the novelty of printing your name on the video screen to wear off. There are more interesting ways of using the TRS-80, including the manipulation of variables.

A variable is a code for something, which might be your name, your driver's license number, or any other piece of information you choose. The fact that you can change this code any time makes it variable, but once you've defined it, the computer will make use of this code any time you instruct it to.

String Variables

Using the methods you learned last time, type enter and run the program in Listing 1. There are a few new points to make here. First is the use of the dollar sign after the letter N. The letter is being used as a variable, but the dollar sign makes it a particular type of variable called a string variable. A string is simply a collection of the characters we would normally place between quotes for a PRINT command.

In Listing 1, N$ (pronounced EN-STRING) is a string variable which we have declared as a code for the words, "THIS IS A STRING, 1,2,3, TESTING". Each time we ask N$, that's what we get.

You may not realize it yet, but this is a mighty powerful instruction. It means, for example, that you can print a phrase of up to 240 characters or so just by using the command PRINT N$. Even better, the Level II machine lets you use many string variables.

You can use each letter of the alphabet, a letter and a number (A1$, B3$ and so on) or two letters (AZ$, BD$), as long as the string sign is used to instruct the computer that this is a string variable. If you leave out the string sign, the computer will normally reject any attempt to equate the variable code letter to a string of letters, because a letter with no string sign means that the variable is a number.

"These statements are called variable assignments and the equality sign doesn't mean equals when it's used this way, but rather 'takes the value of.'"

The exception occurs at the very start of a program, when you have told the computer that all variables which start with a specified letter will be string variables. This is done by using the DEFSTR (define as string) command. A program starting with 10 DEFSTR A,K,T uses variables such as A, AM, AA, K1, KZ, TZ, TT and so on without the string sign after them.

Let's look a bit harder at this string thing. Can we really store a long string? Try Listing 1 again, but this time make the first line read:

```
10 N$ = "THIS IS A MUCH LONGER STRING WHICH
WILL NEED MORE MEMORY SPACE THAN THE
PREVIOUS ONE"
```

Don't change the remaining lines, but type in the new line 10, ENTER it and RUN.

The Clear Command

So, you got an error message? Even if you typed everything correctly and didn't get the SN error message, you still get the words OS IN 10. OS means out of string space; was Sinclair wrong? Something else you have to learn about the TRS-80 is you have to let it know in advance how much memory it needs to reserve for strings.

Normally, when you first switch on, the 80 reserves 50 units (bytes) of memory for strings. Each character of a string, and that includes spaces, remember, takes up one byte of memory, so you don't need to have very long strings to total over 50 characters. To reserve more space, use the CLEAR command at the start of a program. Try it by typing in the following line at the beginning of the program:

```
5 CLEAR 200
```

Leave line 10 as it is; ENTER and RUN. This time there should be no OS error message, because we've reserved enough string space for 200 characters. Now this may seem confusing, because when you are inventing a program you may not know just how much string space you need. That's O.K., because you don't have to enter the CLEAR instruction until your program is complete and ready to RUN, and by that time you should be able to tell how many characters are going to be stored as strings. If you forget, it's no great hassle to type in a line 5 with a CLEAR instruction, followed by a number big enough to

store all your characters. Lines are numbered in tens in order to leave room for second thoughts like this.

Why should we have to do this? Well, it's all tied up with the way the computer controls the memory space. We said in the first article of this series that it is possible to reserve space at the top of memory for machine-code programs.

This is not the only reserved space in the memory. The memory space just below the machine-code space is reserved for strings. If you haven't used a CLEAR (number) instruction, only 50 bytes of this memory are reserved. Use more than 50 bytes of string, and you get the OS warning, because you have run out of reserved space, and that part of memory is in danger of being used for something else.

Why don't we just start every program with CLEAR 2000, reserving plenty of space? Simple: It's wasteful. Reserve too much space in memory, and it's like roping off half a parking lot—you're wasting space. Memory is valuable to the computer, so we don't reserve any more than we need, especially when we're entering a long program.

The way computers use strings (called string handling) is one of the points that sets apart the serious computer from the "just fun" machine. It's the big, big improvement of the Level II machine over the Level I, for example.

The little program that we've been running gives you a taste of this. In line 40, the PRINT command asks for a print (on the video screen) of the message we've coded as N$, but also for the message ";ALL WELL." Notice the positions of the quotes and the semicolons? The semicolon immediately after N$ is a command, meaning put in a space and keep printing on the same line. The semicolon inside the quotes is part of the message and it gets printed. There's nothing to show, when you look at the whole message on the video screen, that one group of characters was stored as a string and the other as a PRINT command inside quotes.

Here we should mention the matter of numbers (more in Part 5, incidentally). If a variable letter isn't specified as a string by the dollar sign or the DEFSTR command, then it's a number. We'll find later on that we can define three types of numbers, but for the moment we won't look for complications. We can write a line, such as 20 A = 15, and then throughout the program we can use A instead of having to type 15. If we want to change it, we use another statement, such as 100 A = 16. These statements are called variable assignments, and the equality sign doesn't mean equals when it's used in this way, but rather "takes the value of."

This is very important, as you'll see later, because some statements look odd if you assume that = means "equals." Take a look at the short program in Listing 2. N$ is a string variable which we set to be "GREEN BOTTLES" in line 10. The number variable A is set to 10 in line 20. When we get to line 30, we get... well, try it for yourself! If we now add a new line:

```
35 A = A - 1 : GOTO 30
```

and RUN again, we see some wild printouts which won't stop until we press the BREAK key.

What happened? We did say that = means "takes the value of." In line 20, A takes the value of 10, so in line 30 you get:

```
10 GREEN BOTTLES, HANGING ON THE WALL
```

(You did get the comma *inside* the quotes, didn't you?) At the new line 35, A takes the value of 10 – 1, which is 9. The colon marks a new instruction on the same line. This saves us from having to make a new line number. The next instruction is GOTO 30—go back and carry out the instruction in line 30 and go on from there. This is the PRINT instruction all over again, so you get:

```
9 GREEN BOTTLES, HANGING ON THE WALL
```

The program then automatically steps to the next line, 35 again. This time A starts at 9; the instruction A = A – 1 gives A the new value of 8 and so on. This is called a loop—the program simply goes from instruction 30 to 35, then back to 30 again, and you can't get out of it except by pressing the BREAK key, by another program instruction or by letting it run out of numbers.

The INPUT Instruction

We need to look now at a more immediate way of entering information into the computer. So far, every string and number we've used has been planned ahead and put into the program from the beginning. The only method we have of changing things is by re-typing the program lines (I'll talk about editing them later).

The instruction that saves us a lot of time is called INPUT, and an example of its usage is in Listing 3. Type in the program, remembering that the @ sign *must* follow directly after the T of PRINT, no spaces allowed, and the number of the PRINT@ position must be followed by a comma and then the first set of quotes. If you run the program, right away the screen clears, and the words:

```
WHAT IS YOUR NAME
```

appear. On the next line a question mark appears, and the program stops, waiting. It's waiting for you to put in your name, or any other name. You can take your time about typing a name, because the computer waits until you hit ENTER. When you do hit ENTER, your name appears with that famous phrase after it. You can enter any name, or any gibberish at the INPUT step. It will accept numbers, or mixed names and numbers like CONVICT 99, or anything else you put in. They will get printed just as if they had been placed between quotes in a PRINT command.

This is more useful, because it lets you write programs that look a bit more friendly, for a start. The TRS-80 goes further with its input command than some others, in fact, and lets you use INPUT like a PRINT statement, so you can write a line such as:

```
20 INPUT "WHAT IS YOUR NAME"; N$ : CLS
```

to replace line 20 and 30 in Listing 3. Do I hear an objection? It's true that when you use INPUT to print like this, you can't place the printing where you want it, because you can't have INPUT TAB or INPUT@. Try this for line 20:

```
20 PRINT@22,;:INPUT "WHAT IS YOUR NAME";N$:CLS
```

Watch the sequence of delimiter markings in this one—after the 22 we have comma, semicolon and then colon marks. Notice we don't use a question mark after NAME, but you'll see one when the program runs, because it forms part of the reply to the INPUT command.

Suppose you try to use N instead of N$ after INPUT? You can't do it, unless what you enter is simply a number. If you specify a string variable, you can INPUT what you like, up to 255 characters; but if you specify a number, then you must enter a number, no letters permitted.

Using INPUT statements to make a sort of conversation is illustrated in Listing 4. In line 20, your name is assigned to N$ by the INPUT statement, and line 30 makes a friendly comment.

At line 40, the INPUT asks for age, and at line 50 for this year. The grand finale is in line 60, when the printout on the video screen gives the name and year of birth. How? Since it has the present year, represented by variable Y, and your age, variable A, it only has to subtract A from Y to get your year of birth—unless you lied about your age! Simple—but it looks like magic to anyone who hasn't seen your TRS-80 in action before.

CLOAD and Friends

CLOAD is one of the instructions we use many times on the TRS-80. It means Cassette Load, and it's the instruction that lets you use these programs on cassette.

The freedom that cassette loading and saving gives you is immense. Without cassettes, each program you use is lost whenever you type a new program or switch off. By saving your programs on cassette you can enter them at any time.

In addition, cassettes give you a chance to run programs which might take many hours to enter from the keyboard, or which most of us could never devise even if we were locked in a padded cell for a year.

O.K. let's go over cassette loading in detail. If

```
5 REM FIG.2.1 INTO 80'S
10 N$="THIS IS A STRING , 1,2,3, TESTING"
20 PRINT N$
30 PRINT
40 PRINT N$;"; ALL WELL"
```

Listing 1.

```
5 REM FIG 2.2 INTO 80'S
10 N$="GREEN BOTTLES"
20 A=10
30 PRINT A;;N$;" ,HANGING ON A WALL"
40 END
```

Listing 2.

```
5 REM FIG.2.3 INTO 80'S
10 CLS
20 PRINT@23,"WHAT IS YOUR NAME?"
30 INPUT N$: CLS
40 PRINT@17,N$;" -THIS IS YOUR LIFE!!"
50 END
```

Listing 3.

you bought a complete TRS-80 outfit, you'll have the CTR-80 recorder; a used TRS-80 might come with a CTR-41. I use a fairly high-grade recorder which has a better-than-normal frequency response (which means it records and plays high notes better). This is advantageous, because the recording and replaying of data and programs use high notes, and you can't load properly unless these notes are loud and clear. For example, if the tone control of a cassette recorder is set to reduce high notes, it just won't load programs.

Whatever cassette recorder you use must have a microphone input socket, automatic recording level adjustment, an earpiece output socket and a motor control. The motor control takes a small (2.5mm) jack plug, the smallest plug on the TRS-80 cable, which comes from the cassette outlet on the TRS-80 keyboard.

The signals out of the cassette recorder come from the earpiece socket, a 3.5mm one, which is linked by pushing in the black plug at the end of the TRS-80 cassette cable. For loading cassettes, you don't need the microphone plug (the grey one), but it looks neater if you put it in place, so in it goes to the microphone input of the recorder. Check them all again.

If you are not using the CTR-41 or CTR-80 recorders, then whatever you use must have the same plug-in arrangements, particularly the motor control, because the recorder motor is controlled by the computer. If you bought only the TRS-80 keyboard and are using your own cassette recorder (or reel-to-reel recorder), then you will have to make or buy adaptor leads.

If you are using the CTR-80 from Radio Shack, then the recorder will run fast forward or fast reverse even when the computer has stopped the motor from running in the play or record/play settings. If you are using the CTR-41 or any other cassette recorder, you won't have this rather useful facility. For some program work, it won't matter, but if you would like to go from one place on the tape to another, then there are various fixes. A few user-group magazines will show you how to cut tracks inside the recorder to do this.

My own fix is illustrated in Fig. 1. It consists of an adaptor box and a small switch which allows either normal or computer control of the recorder motor. With this addition, you can also use manual control with the cassette recorder switched to play, which is useful for finding a short gap between the programs on the tape. If you start a playback in the wrong place on the tape, it won't load correctly and the program won't run. If hardware doesn't interest you, the easy solution is to type:

```
10000 OUT 255,4: GOTO 10000
```

ENTER this and RUN, and the motor will stay switched on by the computer until you press BREAK.

Loading the Program

So we've sorted out our recorder, everything's plugged in, and we're ready to go. Next, we need a BASIC program on a cassette. My TRS-80 came with Radio Shack's blackjack program, and it's likely that yours did too. If not, then you'll need some software.

```
5 REM FIG.2.4 INTO 80'S
10 CLS
20 INPUT "WHAT IS YOUR NAME, PLEASE";N$
30 PRINT:PRINT N$;" I LIKE THE SOUND OF THAT"
40 INPUT "TELL ME PLEASE WHAT AGE YOU WILL BE THIS YEAR
 (IN WHOLE YEARS)";A
50 INPUT"AND NOW WHAT YEAR THIS IS";Y
60 PRINT:PRINT"SO, ";N$;" ,YOU WERE BORN IN ";Y-A
70 END
```

Listing 4.

Pop the cassette into the recorder, with the program you want to load, so that the label of the wanted program is uppermost. Rewind the tape completely—the CTR-80 will make a moaning noise when the rewind is complete.

Set the volume control of the recorder to halfway between its maximum and its minimum settings. Make sure that the tone control, if you have one, is set to give maximum treble.

Now we're ready. Type CLOAD on the TRS-80 keyboard, press play on the recorder and press ENTER on the keyboard. You should hear the motor of the recorder start to hum. If the motor starts when you press play there's a fault in the motor circuit somewhere. The motor-control jack may not be plugged fully in. If the motor doesn't start at all, then perhaps there are no batteries or the power line isn't plugged in. These are what we call hardware problems.

Another possible hang-up could be a software one. Are you sure that you typed CLOAD? Keybounce, which may have given you CCLOAD or CLLOAD won't be accepted by the computer, and it will snap back with an SN error when you hit ENTER.

By now, if all has gone well, the cassette should be running. Unless you have connected the loudspeaker of the cassette recorder so that you can listen to the tape as it plays, you won't know when the action actually starts, until you see things happening in the top right-hand corner of the monitor.

Two asterisks appear once the program starts loading, one steady, the other flashing slowly. One asterisk flashes at the rate of loading program lines, on for one line, off for the next. If you're loading a short program with short program lines, the rate of flashing will be rapid, and it won't be long before a click comes from inside the keyboard unit, the cassette recorder motor stops, and READY appears on the video screen.

If all this happens, you have achieved a successful load first time, and you qualify for the Fort Worth Perfboard Medal of Honor.

It's much more likely, first time round, that things won't run quite so smoothly. There are two extremes to the problem. One is that no asterisks appear at all. This could simply be due to a tape which starts only after a long leader. which is why it is so useful to have a loudspeaker tone; but if there is no trace of the asterisks after a minute, there is replay volume trouble. Despite what the manual may say, this indicates that the replay volume is either much too low or much too high. If, on the other hand, you get two asterisks, but the right-hand one isn't flashing, then it's a dollar to a cent that the replay volume is just a little bit too high.

Cassette Control

If you have either of these problems, you'll soon find that you have another one as well,the cassette recorder motor keeps humming away happily until it comes to the end of the cassette, or until you do something about it. It certainly won't stop at the end of the program load, because the stop instruction was never loaded into the computer. You can waste a lot of time just waiting for a cassette to load, so keep a careful eye on these asterisks. If they aren't blinking properly, then press the RESET button at the back of the computer, rewind the cassette, press CLEAR to remove the old instructions and asterisks and start again with a different volume control setting.

Don't give up if you overshoot and go far too high or far too low. When I bought my first TRS-80, I spent the better part of an afternoon trying to achieve a good load. Since finding the correct settings, it has never at any time failed to load a good cassette. You don't need to use expensive chrome-dioxide tape material, just reasonably good quality audio tape, like Agfa or TDK. It's definitely an advantage to use tape sold in short lengths for computer work, but the C60 length is very useful when you're developing a program with several versions.

You may find that you have a tape which simply won't load under any conditions. The odds are that it's a tape intended for a Level I machine. Once again, if you have the sound wired on your cassette recorder, you can check this, because the Level I tapes have a lower pitched note and sound quite different.

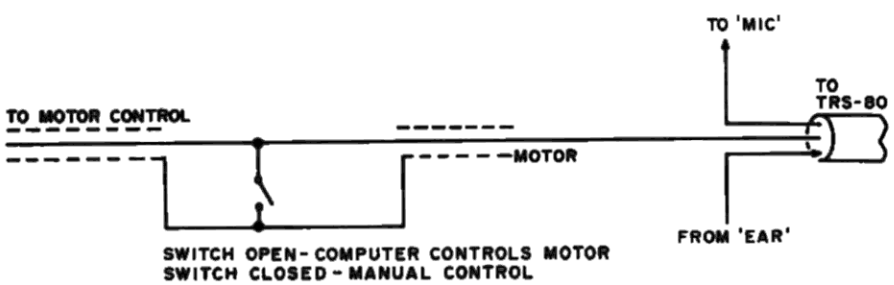


Fig. 1.

✓Reader Service—see page 226

Once you have found the correct setting for the Radio Shack blackjack tape, or whatever you use for trials, try to find the limits of the volume control settings. It takes a lot of patience, but it's worth your while on the older models (later models are more tolerant of volume control settings). Load the cassette at different settings checking only for a few seconds that the asterisks are flashing correctly, then use RESET to stop the tape and rewind. You should end up with two marks on the volume control, one at the lowest position at which a tape will load, one at the highest.

Set the volume control for normal operation midway between your marks. If you then find a tape doesn't load correctly at this midway setting, try it at each of the extreme settings. If it won't load on any of your marked range, reject it. One final check: Make sure it is a BASIC tape and not a machine code (system) tape, which requires quite a different technique (coming up).

When you have loaded your BASIC tape, type LIST and hit ENTER. The program will now list, unless it's one which has been specifically coded to prevent copying. As the program lists, it will scroll up the screen fairly rapidly.

You can stop the scrolling any time by hitting SHIFT and @ at the same time, so I usually keep one finger on the shift key and another on the@ key to stop and start the scrolling. Any other key can be pressed to restart scrolling.

I didn't say what scrolling means? When you've seen it, you'll know—it's the way the listed lines appear at the bottom of the screen, seeming to push previously listed lines off the top of the screen.

What you should be looking for in the listed program is corruption—gibberish lines, sometimes with no numbers or numbers out of order. Trouble is, until you get to know a bit more about programming, you don't really know a strange line from a perfectly good one! The real test, of course, is to run the program. If it operates perfectly, then there is nothing wrong with your cassette recorder volume control setting, and you can look forward to a long, active computing life.

Just keep your head clean. I mean, of course, the record/play head of the cassette recorder. Get a pack of cleaning fluid (isopropyl alcohol) and cleaning pads and use them as per the instructions every three months or so, depending on how much tape you use. Don't be too generous with the fluid, as it can sometimes swell the plastic bearings inside cassette recorder motors.

Loading System Tapes

While we're on the subject of this cassette loading caper, we might as well look at how machine code tapes are loaded. A machine code, or system tape usually comes with a bit more information than a BASIC program tape.

For one thing, you'll need an answer to the MEMORY SIZE question which appears when you switch on. This is a number, such as 32000, that reserves some memory. The system tape, or the instruction sheet which comes with it, should have the correct number printed on it. If the program is one which doesn't need reserved memory, or which reserves its own, the instructions will say so.

Hit ENTER and the usual Radio Shack

message comes up, with READY. Next, type SYSTEM, and hit ENTER again. This time, you'll get an asterisk and a query at the left-hand side of the video display. That's 80 language for, "What's the code name for the tape?"

The code name will have up to six letters and must be typed. For example, the Radio Shack fix tape for keybounce has the code name KBFIX. When you've typed the name, prepare the cassette recorder to replay, press the play key and hit ENTER. The cassette recorder motor will start, and, if all is well, you should see the usual asterisks to indicate that you are loading your first machine code program. The rate of flashing is usually a lot slower than it is for a BASIC program, so don't worry if the load stops after only a few slow flashes.

Now what can go wrong? Well for one thing, the code name which you typed may not be the code for the first program on the tape. If it isn't, the left-hand asterisk will be replaced by a letter. If that letter is C, then you have trouble, and you'll have to try again with a different volume setting. Hit the RESET switch to stop the action, rewind the tape, clear the screen and start again by typing SYSTEM. You don't have to switch off and answer the MEMORY SIZE question again.

Next question: Having loaded it, how do you run it? When the tape has finished loading, the recorder motor cuts out, the asterisk stops flashing, and another asterisk and query appear under the first one on the left-hand side of the monitor. Type a slash (/) and then the entry number of the machine code program.

Machine code programs are not so simple as BASIC programs in this respect: You have to instruct the computer where to start working. The entry number should, once again, be noted either on the cassette or on the instruction leaflet. It may be the same as the number you used to answer the MEMORY SIZE question. When you've typed the slash and the number, hit ENTER and your machine code program will run.

Suppose you quit using one machine code program and want to start using another one which needs more memory roped off? You don't have to switch off to do this. Just type SYSTEM, hit ENTER and when the asterisk and query appear, type slash and 0 and hit ENTER again. The MEMORY SIZE question will appear again. You'll lose any BASIC programs you had in store, though, so if you have mixed BASIC and machine code, make sure that you have the BASIC program on tape.

A few machine-code programs are "self-locating." Once you have loaded them in by typing their code names and entering, the second step is just to type the slash and hit ENTER. Whatever type of machine-code program you may be using, don't forget the slash. Otherwise, you'll find that when you hit ENTER, the cassette motor starts running again, trying to enter another program, and you'll have to recover control by using the RESET button. You'll probably lose the program which was loaded, but you can start again.

Recording Programs

We've left until the end the matter of recording BASIC programs of your own. You'll want to record your own programs, of course, to remind

yourself how good you are. You'll also want to make back-up copies of software you've bought, just in case anything should happen.

How do you record a program? The first step is to prepare a blank cassette. Don't think you can re-record an old tape in the same cheerful way you may be used to with audio cassettes. You might get away with it, but odds are you won't, and your recording will be corrupted. If you want to re-record a tape, wipe it completely with a bulk eraser. If the program you want to record has taken you a long time to run correctly, you won't want to trust it to anything but a length of good quality fresh tape.

Reel the cassette back to the start and take a look at it. If there's a leader, a piece of clear or colored plastic tape at the beginning, advance the tape a bit until the grey magnetic coating is visible. I usually run each tape for a count of five on the tape-footage counter. Don't touch the tape; it will leave a greasy mark which can cause loading problems later. Place the tape in the machine again, note the counter setting and press the record and play keys. A few cassette recorders use one single record key, but most use the safer system of needing two keys for recording.

The volume control setting doesn't matter, because recording volume is automatically controlled, unlike replay. Now type CSAVE and a quote mark, then a letter and another quote. If you choose "A" as the letter, this will appear on the screen as CSAVE "A".

If you don't use the code letter, the computer will reject your attempt to record, but only after it has already recorded a signal on some of the tape, which you won't be able to use again, unless you can erase it thoroughly.

When you're satisfied that all is well, hit ENTER, and the program should start to record. There are no flashing asterisks to remind you this time, just the quiet hum of the motor of the cassette recorder until it clicks off at the end of the recording. The click, incidentally, comes from the relay inside the TRS-80. At the end of the CSAVE, READY appears on the screen.

At this point, don't start shouting eureka and running around. You don't know yet that you have a good recording. Rewind the tape, type CLOAD?"A" (or whatever letter you used) and press play on the recorder. Then check again that the query mark has been typed after CLOAD; hit ENTER and wait. The program will play back, with the usual flashing asterisks, but this time the replayed program is being compared, byte by byte, with the program which is still in the memory of the computer.

If they aren't identical, the mesage "BAD" will be displayed. You have then to sort out whether the tape copy is faulty, or you need a different volume control setting for this program. Only when you've CSAVEd and CLOADed with no error messages can you be sure that you have a good copy of your program. Cautious people always make two recordings, checking one with CLOAD? People like me who shed blood, sweat and tears to create a program always make three copies.

Be very careful that when you use the CLOAD? command, you don't leave out the query mark. If you do, the program on tape will load, replacing the program that was in the computer. If the recording was good, this won't matter, but if the recording was bad, you have lost the good original and have a bad copy, and that just isn't fair trading.

The CSAVE Instruction

Very little ever seems to go wrong with a CSAVE instruction, but there are a few points you will need to remember. One is that the computer can only control the motor of the cassette recorder; it has no control over the rest of the recorder.

If, for example, you use CSAVE but forget to press the record and play keys of the recorder, or press only one of them, then the computer will push out the recording just the same, with no warnings and no recording made. It might be useful to arrange it so that you got an error message, but this would need more connections between the computer and the recorder and would make the recorder a non-standard item.

You should always use CLOAD? after a CSAVE, so you can check that you really did record that program. A much worse fault is to type CLOAD and run with the record and play keys down. This way you load no program, and you wipe out anything which was on the tape!

When you CSAVE a program, you have to use a letter or a couple of letters of letter/number—it's like choosing a name for a variable. If you don't, as we've said, the CSAVE will not run, an SN error will be displayed, and the tape will be corrupted.

The label is called a filename, and it's important to the recording. It's used when you CLOAD the program, and it's particularly useful when you have several short programs packed together on a piece of tape. Suppose you have three programs on the start of a C15 cassette, and they have been labelled "A", "B" and "C" at the time they were CSAVEd.

When you CLOAD, you can type CLOAD?"B" and hit ENTER, and start running the cassette from the start. When the first program starts to replay, the left-hand asterisk will be replaced by the letter A to show you that this is the filename (the first letter if there's more than one) of the program which is being read. The other asterisk will flash normally. When the program which you have requested comes on line, it will load in the usual way, with one steady asterisk and one flashing one, then the recorder will switch off.

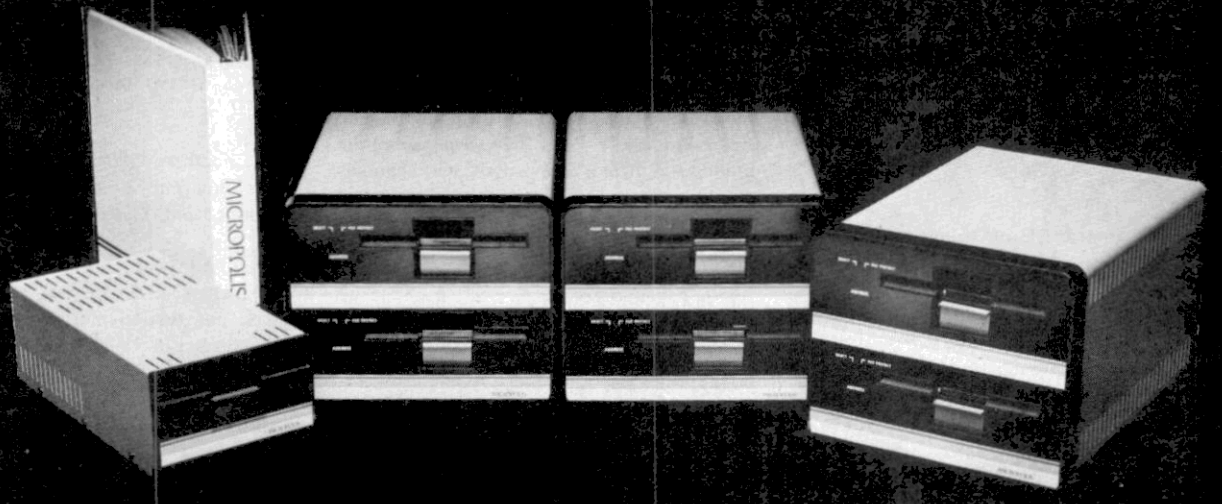
Normally, when I keep several programs on one cassette, I leave plenty of space between and use the tape counter to find each one, but I find this "label-search" very useful for my backup cassette, which is a C60 with all my most valuable programs stored tightly together. Since I use this only when a valuable program has been wiped or corrupted (and I'm resting the *other* backup cassette), it doesn't matter if it takes twenty minutes to find the program.

One last point—always start a replay either at the start of a cassette or at a point where you know there's no program recorded. If you start running where there's a program recorded, the load will be faulty, and the computer can lose control of the motor. You'll end up having to use the RESET button and rewinding the tape. ■

LESS THAN SUGART • PERTEC • MPI

GIANT MICROPOLIS SALE

For TRS-80 only
Absolutely ends October 31, 1980



MICROPOLIS 35 TRACK 5¼" DRIVE 89.6 Kb capacity	**$358.66** Reg. $545.00
MICROPOLIS DUAL 35 TRACK 5¼" DRIVE 179.2 Kb capacity	**$716.43** Reg. $1,036.00
MICROPOLIS 77 TRACK 5¼" DRIVE 197.12 Kb capacity	**$442.71** Reg. $675.00
MICROPOLIS DUAL 77 TRACK 5¼" DRIVE 394.24 Kb capacity	**$808.56** Reg. $1,299.00
NEWDOS/80 and 10 FREE DISKETTES	**$149.00**

Alpha Byte Storage

4636 Park Granada
Suite 159
Calabasas, California 91302
(213) 883-8594

✓69

• Free shipping included • All drives are brand new from the factory • Mail or phone orders accepted • We take Master Charge or Visa • Dealer orders accepted • Offer good as supply lasts • See our other ad in this months publication for more great deals • Send for our free giant 50 page catalog.

All drives are brand new and include chassis and power supply.

Let's look at the management instructions involved and how to use them effectively.

Pulling Strings Together

John D. Adams
13126 Tripoli Ave.
Sylmar, CA 91342

My first article on strings outlined their concept. The following examines string management instructions.

The TRS-80 handles string and non-string data in different ways. These two types of data are not interchangeable on Level I systems.

Level II, however, gives us a way to get around the problem with two instructions: VAL(n$) and STR$(n).

VAL(n$) and STR$(n)

These two instructions permit us to use data stored as strings in non-string operation and vice versa. The instructions are quite easy to understand. Let us assume that we have "12345" stored in location A$ and "54321" stored in location B$, and that during the execution of a program we need the sum of the two numbers. Enter and RUN the following:

```
10 READ A$,B$
20 DATA 12345,54321
30 PRINT A$ + B$
```

Note what line 30 produces. Since the numbers are in memory as symbols rather than values, the computer did not return their sum, but their concatenation. Is there a way to get a sum from these numbers? Change line 30 to read:

```
30 PRINT VAL(A$) + VAL(B$)
```

This produces the needed sum. The VAL statement has instructed the computer to use the *values* of the strings. By including the line C = VAL(A$):D = VAL(B$), we store the numbers in memory as values C and D and as strings in A$ and B$.

STR$(n) accomplishes the reverse. Should there be a number in memory as a value and it is needed for use as a string or as part of a string, this instruction converts it. Numbers can be converted either way, that is, from symbol to value or from value to symbol, but we cannot do the same thing with letters. RUN the following lines:

```
10 A$ = "ABC":B$ = "ABC123":C$ = "123ABC"
20 PRINT VAL(A$)
30 PRINT VAL(B$)
40 PRINT VAL(C$)
```

The first two lines of the printout return zeros, since they start with letters. Line 40, however, prints the numerical portion of the string. Letters do not have value, so they are ignored by the instruction. (Letters do have ASCII code numbers, but they are for identification only and have nothing to do with numerical value.) This feature distinguishes between strings starting with numbers and those starting with letters, as in the following lines:

```
10 INPUT"ENTER ANY STRING, EITHER NUMBERS
   OR LETTERS";A$:B = VAL(A$)
20 IF B=0 THEN 30 ELSE PRINT B:GOTO 10
30 PRINT"STRING STARTS WITH
   LETTER - CANNOT RETURN VALUE":GOTO 10
```

STR$(n) works the opposite way. Try these lines:

```
10 A = 1970:B = 1980:A$ = "TOTALS FOR":B$ = " TO"
20 C$ = A$ + STR$(A) + B$ + STR$(B)
30 PRINT C$
```

The numbers 1970 and 1980 are stored as values, but line 20 allows them to be incorporated into C$, using the STR$ instruction.

The LEN(n$) instruction counts the number of characters in a specified string, including spaces, punctuation marks, symbols, etc. It counts leading or trailing spaces *only* if they are included as part of the string. We have to watch out for this when numbers have been converted to string data by the STR$ instruction. These lines illustrate the problem:

```
10 A = 12345:B$ = "12345"
20 A$ = STR$(A)
30 PRINT LEN(A$)
40 PRINT LEN(B$)
```

Why do we get a different count for the two strings? Remember that when the TRS-80 prints a number, it always leaves one leading space for the sign, whether it is needed or not. This space explains the greater count, as it was transferred to the string. We will put this instruction to use after we have looked at the three statements which follow.

LEFT$(n$,n), RIGHT$(n$,n), MID$(n$,n,n)

These functions can be used to "excerpt" a string. Using them, we can pick up any portion of an existing string to use elsewhere. Each of them has information, called the "argument," enclosed in parentheses. In the LEFT$ instruction, the first term of the argument states the name of the string we want to excerpt, followed by a comma. The second term indicates the number of characters we want picked up, start-

ing with the leftmost character. An example follows:

```
10 A$ = "12345"
20 PRINT LEFT$(A$,1)
30 PRINT LEFT$(A$,2)
40 PRINT LEFT$(A$,3)
50 PRINT LEFT$(A$,4)
60 PRINT LEFT$(A$,5)
```

The printout illustrates what LEFT$ does. RIGHT$ does the same thing, but starts counting backwards from the rightmost character. Add the following lines:

```
70 PRINT RIGHT$(A$,1)
80 PRINT RIGHT$(A$,2)
90 PRINT RIGHT$(A$,3)
100 PRINT RIGHT$(A$,4)
110 PRINT RIGHT$(A$,5)
```

The printout shows that we can use these two statements to "pick off" any desired number of characters from either the beginning or the end of a string. In both of the instructions, the second term of the argument (number of characters) may be a number or a variable. If, for example, the number four is stored in memory location X the statement LEFT$(B$,X) will excerpt the first four characters of the string in location B$. Should the second term of the argument be larger than the number of characters in the string, the entire string will be returned.

The third statement, MID$(n$,n,n), excerpts portions from the middle of an existing string. There are three terms in this argument. The first indicates the string to be used, the second represents the position at which the lift is to start, and the third indicates the number of characters to be lifted from that starting point.

MID$(A$,12,7) returns seven characters from string A$, starting at position 12. The second and third terms may be variables such as MID$(L$,X,Y) in which numbers stored in X and Y determine the starting point and the number of characters to be returned. A simple routine is given in Listing 1 to illustrate the use of these three instructions. It also uses the LEN(n$) instruction.

Line 10 clears the screen, defines variables and deposits a comma in string location A. Lines 20-40 request your name and, using concatenation, build the string in location B (shown in line 45).

The number of characters in the string is now counted by the LEN(B$), and that number is stored in location X. Lines 50 and 60 set up a FOR-NEXT loop which, using the MID$ instruction, examines each character in the string starting at position one, character one, and continues until it finds the period after the middle initial.

At this point, execution proceeds to line 70. Here the RIGHT$ instruction stores the last name in E1 and then line 80 uses the LEFT$ instruction to store the first name and middle initial in E2. Line 90 prints out the results. These statements allow almost unlimited flexibility in the construction and use of strings. Experiment with them a little.

The FRE(n$) instruction may be used either in the command or the execute mode and returns the amount of string storage space available at that point. It requires an argument in parentheses, but the argument is what is called a "dummy" argument. To get an idea of how this works, load and run the routine in Listing 1. After the printout, type FRE(A), hit ENTER, and the computer will return the number of string space bytes left after entering your name.

The argument A is a dummy; you get the same return if you enter B, C or any other string variable name, even if it is not in use in the program. Try using different variables. If you use a non-string variable in the argument, the computer returns the number of bytes left in RAM. Enter FRE(X) as an example.

This instruction is very useful in building programs where there are a lot of strings, and you want to keep track of how much space is left. This routine is an example:

```
10 CLS:CLEAR 100:DEFSTR A,B
20 FOR I = 1 TO 20
30 INPUT"ENTER NAME":A(I)
40 PRINT"YOU NOW
   HAVE";FRE(A);"BYTES
   OF STRING SPACE LEFT"
50 NEXT
```

STRING$ (n,character)

Useful in graphics applications, we can instantly create strings of up to 255 repeated characters with this statement. Any letter, digit or symbol on the keyboard may be used, although the ASCII code numbers must be used for the quote mark, comma and colon. The first term of the argument sets the number of characters wanted in the string, and the second term indicates the character itself, or the string location in which that character is stored.

- To print a specific character, use the form PRINT STRING$(50,"*"). Here the desired character must be enclosed in quotation marks.
- To print a character stored in a string location, use the form PRINT STRING$(50,A$). The variable location name (A$) is not enclosed in quotes. That character must, of course, have been previously stored in A$.
- To print a character using its ASCII code number, use the form PRINT STRING$ (50,58). As 58 is the ASCII number for the colon, this command will print a string of 50 colons. The ASCII code number is not enclosed in quotes. Using these codes, any character may be printed, including the graphics patterns which are ASCII numbers 129 through 191. All of the code numbers are listed on pages C/1 and C/2 of your user's manual.

One of the advantages of using this instruction is its speed of operation and printout. To see the difference, enter and RUN the following lines:

```
10 CLEAR 500:INPUT"ENTER CHARACTER
   TO BE USED";A$:CLS
20 FOR X = 0 TO
   254:PRINTTAB(X)A$;:NEXT
30 PRINT " ":PRINT
40 PRINT STRING$(255,A$):GOTO 10
```

For formatting output, strings may be prepared to make borders, single lines for column totals using the minus sign, double lines for columns using the equal to sign, etc., and then quickly called as subroutines.

ASC(n$) and CHR$(n)

In the first part of this article, string comparisons were discussed. At that time, we took a brief look at the ASCII codes. Level II offers two instructions which allow us to operate with these code numbers, if necessary. The ASC(n$) statement gives you the ASCII code number of the first character of the string that you have specified as the argument. As with other instructions in this group, the argument is enclosed in parentheses. If you want to have the ASCII code of a character returned, and you enter the character manually from the keyboard, it must be in quotes. Entering PRINT ASC("A") returns 65, which is the code number for A. When the character is stored in a string location, the quotes are not used. An example would be:

```
10 A$ = "5"
20 PRINT ASC(A$)
```

Running these two lines returns the number 53 which is the code for the digit 5. All code numbers are returned in decimal notation. CHR$(n) performs the reverse operation. In this case, the argument is the ASCII code, and the character itself is returned. Entering the command PRINT CHR$(91) causes an up arrow to be printed on the video monitor. This is a powerful function, as it permits us to print characters not normally accessible from the keyboard.

We know that we cannot use quotation marks inside a statement to be printed, because the computer interprets the second mark as the end of the line to be printed. Such signs are called delimiters. If the use of quotes is necessary in a printed statement, it may be accessed by using the CHR$ statement. Type the following command.

```
PRINT"THIS WILL ALLOW ";CHR$(34);"QUOTE
MARKS";CHR$(34);" TO BE USED"
```

No spaces are included with this command, and they must be provided as in the above after the word "allow" and before the word "to".

Since the down arrow is used for line feed, the left arrow for backspace and the right arrow for tabbing, we cannot print them in the normal manner. Using their codes, however, which are 92, 93 and 94, respectively, we may print them by using a line such as:

```
PRINT CHR$(92);CHR$(93);CHR$(94)
```

All of the graphics characters may be printed using their ASCII codes. Be careful when using these codes, as numbers 0 through 31 are assigned to various control operations. RUN the following lines.

```
10 PRINT"FIRST LINE"
20 PRINT CHR$(13)
30 PRINT"LINE 20 HAS CAUSED A LINE
   FEED TO BE EXECUTED"
```

These codes are quite useful, especially when working with printers.

INKEY$

The last of the instructions we will look at is INKEY$. As the TRS-80 does not come with joystick-type input devices, this instruction is very useful with real time games, allowing the operator to enter information while the program is running without using the ENTER key.

INKEY$ causes the computer to scan the keyboard many times per second to see if any information has been entered. If data is found, program execution continues; if not, the search continues. When a character is entered, it is scooped up and stored in a string location, and the scanning procedure continues. The following lines illustrate this function:

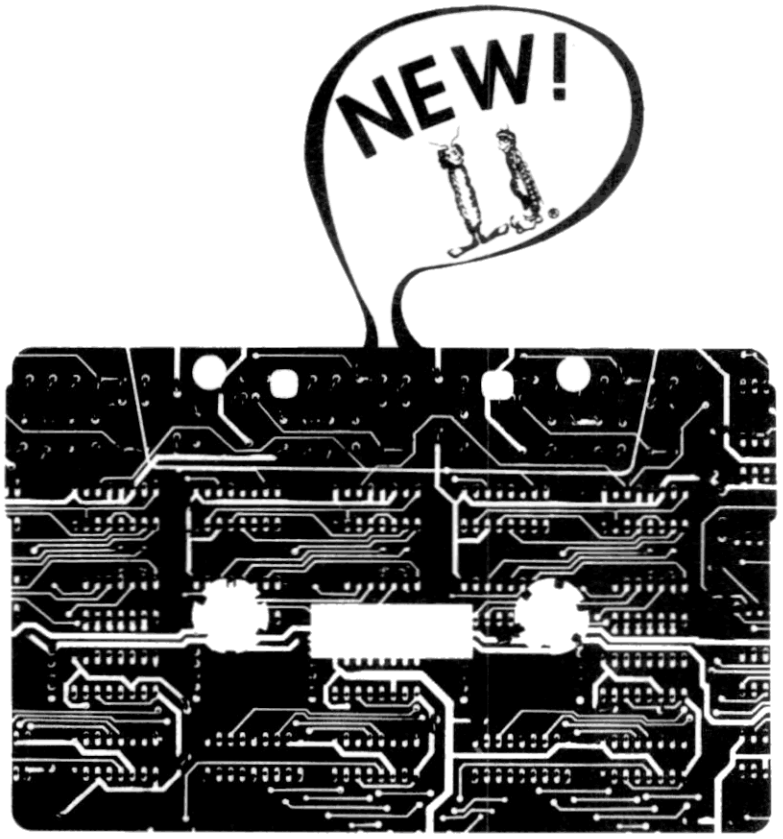
```
10 CLEAR 200:DEFSTR A,B
20 CLS:PRINT"ENTER A MESSAGE. WHEN
   FINISHED ENTER A SLASH BAR"
30 A = INKEY$:IF A = ""THEN 30
40 IF A = "/"THEN 80
50 CLS:PRINT A
60 B = B + A
70 GOTO 30
80 CLS:PRINT B
```

Line 10 clears string space and defines A and B as string variables. Line 20 requests a message and defines the slash bar as the "ending" character. Line 30 sets up the INKEY$ scan. It is a closed loop which continues cycling until something is entered from the keyboard. The empty quote marks ("") in line 30 are used to designate a null string, or one which has no characters in it.

When a key is pressed, line 30 also deposits that character in string location A, and program execution continues. Line 40 tests for the ending character, and if the test fails, line 50 prints the character entered at position 0 on the monitor screen. Lines 60 and 70 then build the string using concatenation.

Execution now returns to line 30 for further entry. When a slash bar is entered, it is detected by line 40, which causes a branch to line 80 for printout and termination. As mentioned, this function bears much investigation to fully realize its potential.

Chapter 5 of the Level II manual starts with the words, "Without string handling capabilities, a computer is just a super-powered calculator." Though this is an exaggeration, using strings makes possible operations which simply cannot be done on a programmable calculator. And the further we dig into these instructions, the more obvious this becomes. ■



✓457

SAMS CIRCUIT DESIGN SOFTWARE.

37 TECHNICAL PROGRAMS TO LET YOU SPEND LESS TIME ON ROUTINE CALCULATIONS, STATISTICS, AND ANALYSES...MORE TIME ON CREATIVE DESIGN AND ENGINEERING.

Sams—the leader in technical publishing—now offers the most advanced engineering software systems available. These tested, documented, debugged programs can be used as stand-alone programs or as subroutines for more complex programs.

■ **FASTER INFORMATION**—Solve simultaneous equations with real and complex coefficients and polynomial roots. Quickly determine the effects of an infinitely variable set of design parameters. Plot graphs for various functions while varying scales to fit data.

■ **EASY TO RUN**—7 cassettes include 37 debugged programs.

■ **FULLY DOCUMENTED**—Easy-to-read and follow instructions.

■ **DESIGNED FOR POPULAR MICROCOMPUTERS**—Designed for use on TRS-80* systems having Level II BASIC and at least 16K RAM; many of the routines can be adapted to run on other popular computers. Programs will be available soon for Apple and Ohio Scientific.

■ **PROGRAMMED, TESTED, AND DEBUGGED BY HOWARD M. BERLIN**—Howard M. Berlin—an author in the Blacksburg Series—is an electrical engineer with the Chemical Systems Laboratory at Aberdeen Proving Ground, Maryland, and has been adjunct instructor in the Department of Electrical Engineering at the University of Delaware.

☐ **PLOTTING GRAPHS FOR VIDEO DISPLAY** (5 Programs)
Histograms. Cartesian Plots. Semi-Logarithmic Plots. Log-Log Plots. Polar Plots. #26006. $16.95.

☐ **PLOTTING GRAPHS FOR LINE PRINTER** (3 Programs)
Cartesian Plots. Semi-Logarithmic Plots. Polar Plots. #26000. $16.95.

☐ **ACTIVE FILTER DESIGN** (6 Programs)
Low and High-Pass (Bessel, Butterworth, 1, 2, and 3-dB Chebyshev). State-Variable Filter. Bandpass Filters with Q's less than 10 and 50. Staggered-Tuned Butterworth Bandpass Filters (2, 3, or 5 states). Notch Filter. #26001. $21.95.

☐ **DESCRIPTIVE STATISTICS & REGRESSION ANALYSIS** (3 Programs)
Descriptive Statistics (mean, standard deviation, variance, kurtosis, z-scores). Curvilinear Regression (linear, inverse, polynominal, exponential, logarithmic). Multivariable Linear Regression. #26002. $21.95.

☐ **ELECTRONICS I** (5 Programs)
Zener Diode Voltage Regulator Design. 555 Timer Design (monostable and astable circuits). Transistor Bias Parameters. Single-Stage Transistor Amplifier Design. Heat Sink Selection and Design. #26003. $16.95.

☐ **ELECTRONICS II** (7 Programs)
4-Quadrant Arctangent Function. Rectangular/Polar Conversion and Complex Number Mathematics. Minimum and Maximum Values of an Array. Roots of Polynomials with Real Coefficients. Inverse Laplace Transforms of a Transfer Function. Solution of Simultaneous Equations with Real and Complex Coefficients. #26004. $16.95.

☐ **ELECTRONICS III** (8 programs)
Average and RMS Values of a Periodic Function. Fourier Series Expansion of a Period Function. Fourier Transform and Spectrum Plot. Analysis of Damped Oscillations. Impedance Matching Pads. Pi-TEE (delta-wye) Transforms. #26005. $16.95.

☐ **CIRCUIT DESIGN PROGRAMMING BOOK FOR THE TRS-80**
Circuit Design Programs for the TRS-80 by Howard M. Berlin features all of the programs listed above...and more. #21741 **$10.95**

Indicate quantities in the boxes above and return entire ad with order.
Mail to: Howard W. Sams & Co., Inc. • 4300 West 62nd Street
Indianapolis, IN 46206. (317) 298-5400.

AMOUNT OF ORDER: $_____
(add local tax where applicable)

☐ PAYMENT ENCLOSED AD058
☐ CHECK ☐ MONEY ORDER
☐ VISA ☐ MASTER CHARGE Interbank No. _____

Expiration Date _____

Account No. _____

Name (print) _____

Signature _____

Address _____

City _____ State _____ Zip _____

SEND INFORMATION ABOUT PRICE AND AVAILABILITY FOR SAMS SOFTWARE FOR: ☐ APPLE ☐ OHIO SCIENTIFIC

Sams Software Systems are available from major Sams Distributors and Computer Stores. Prices good in U.S.A. only. In Canada, contact Lenbrook Industries, Ltd., Scarborough, Ontario, M1H 1H5, Canada. Offer expires 12/31/80.

*TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corp.

Sams Books.

SBIS®

Small Business Inventory System

Accounting information Order entry Inventory control Invoicing

Westech's *Small Business Inventory System* is a powerful and comprehensive inventory / accounting package for the TRS-80. Order entry, goods on hand, inventory control, inventory replacement and restocking procedures are all organized with the aid of the system.

SBIS will handle credit card and layaway sales, order cancellations and discounted sales. Complete lists of all inventory detailing product type, quantities and item code numbers can be printed whenever required. Information available to the operator of the system includes part numbers, item descriptions, product codes, list prices, costs, average costs, quantities on hand, physical locations, when to re-order and who the vendors are.

Items that are 'lost' can be traced with the aid of a unique search routine. It will maintain data regarding lot quantities, quantity of sales year-to-date, sales for last year, item numbers and when an order is due to be delivered. *SBIS* stores and maintains accurate daily, monthly and yearly sales figures.

Storage capacity? As many as 10,000 different line items may be stored on a TRS-80 Model One having four 77 track drives. Capacity with four 40 track drives is 5,000 line items and 2,200 on just two 40 track drives. The TRS-80 Model Two will handle about 25,000 line items. Sorting and looking for items is very fast, as Westech's *SBIS* utilizes machine language in-memory sort routines. The system requires 32K, two disk drives and a suitable printer (for permanent records).

A complete Model One documentation manual and demonstration disk is available for $35 (which is deductible from the purchase price of $545.00. *SBIS* is also available for the Model Two for only $595.00 - a demonstration disk and Model Two documentation is $40.00 (also deductible from the purchase price of the system).

See your nearest *IJG Authorized Dealer* for more information on this exciting *SBIS* time and money saving system.

PARTIAL LISTING OF PARTICIPATING *IJG COMPUTER SERVICES* RETAIL DEALERS

H & E COMPUTRONICS
10 Drexel Court
Spring Valley, NY 10977
(914) 425-1535 (800) 431-2818

THE DATA CONNECTION
11818 Wilshire Blvd.
Los Angeles, CA 90025
(213) 479-1980 M. McCartney

BIAS, INC.
49-A Cedar St.
Manchester, CT 06040
(203) 643-2544

LINGO ENTERPRISES
1052 Artesia Blvd.
Long Beach, CA 90805
(213) 422-0289

HEINZ WILGEN
Tandy/R.S. Vertragshandler
Im Handenhof 11, 2900 Oldenburg
Germany (0441) 3 68 22

DATA FLOW CORPORATION
24312 Via Viejo
El Toro, CA 92630
(714) 768-7623

THE PROGRAM STORE
4200 Wisconsin Ave., N/W
Washington, DC 20016
(202) 337-4691

LEVEL IV PRODUCTS, INC.
32238 Schoolcraft
Livonia, MI 48154
(313) 525-6200 (800) 521-3305

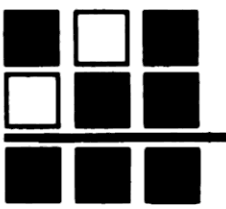
CUSTOM APPLICATION SOFTWARE
P.O. Box 1119
Placentia, CA 92670
(714) 996-7474

HOUSTON MICROCOMPUTER
5313 Bissonnet
Bellaire, TX 77586
(713) 661-2005

DC COMPUTER PRODUCTS
596 Taylor Way
Belmont, CA 94002
(415) 592-4222 (800) 227-7362

APPARAT, INC.
3973 South Olive
Denver, CO 80237
(303) 758 7275

WRITE, PHONE OR CALL FOR FURTHER DETAILS



✓37

IJG COMPUTER SERVICES

569 N. MOUNTAIN • SUITE B • UPLAND, CA 91786 • (714) 982-7829

*TRS-80 is a trademark of the Tandy Corporation.

RADEX

random access data executive T.M.*

The I.J.G. Random Access Data Executive (*RADEX-10*) is a sophisticated database management system. Written in disk BASIC, the program enables you to create and manipulate databases with up to 10,199 records, any of which can be accessed within seconds. The minimum system required to use *RADEX-10* is 32K of memory and 2 disk drives. A printer is not needed to use the program.

RADEX-10 enables you to create and manipulate databases easily and quickly, without any programming. You can use *RADEX-10* to maintain any type of file or record requiring fast access and maximum use of disk space. As a self contained system *RADEX-10* is almost self-explanatory. All operator prompts and messages are in plain english - not computerese.

The standard system consists of six program modules, which load and execute automatically as required. You can create files, add data, remove data, change data, generate reports, produce mailing labels (if you have a printer) and generally manipulate your data - all without writing a single program line!

All of the program modules are designed to handle specific tasks within the system. The modules are designed to be 'transparent' to the user, all file creation and manipulation being taken care of automatically.

The Report module is one of the most flexible and powerful available for a TRS-80 database system. It allows you to search all the records, or a selected range of records, and list only the records that meet the conditions specified. You can specify up to 30 separate conditions that a record must meet, and any of the conditions can be applied to any separate part of a record. Conditions that can be selected are; equal to, greater than, less than or alphanumeric match (on alphanumeric parts of records). Logical operators AND, OR, AND NOT and OR NOT can also be performed on the specified conditions.

After the conditions for a report are specified they are stored on disk, so that you can have several different reports available on the same data.

Reviewed in the July issue of *80 Microcomputing*, *RADEX-10* comes with a 40 page manual and will operate with TRSDOS or NEWDOS. Versions are available for 35, 40 or 77 track disk drives. This extremely versatile system is only $99.00, the manual is available separately for $15.00 (with full credit towards program purchase).

C.A.S. Report Sort Module

This new add-on module for *RADEX-10*, or the *Universal Database Manager*, allows you to sort your *RADEX-10* database on any field and output the result to a printer or the screen. The report will also be sorted on any specified output field. A high speed in-memory machine language subroutine, especially developed by *RACET computes*, is used to perform the sorts.

C.A.S. Database Sort Module

Using this module you can sort your entire *RADEX-10* database, or portion of the database, on any field. The sorted records can be output to a new file, or appended to the existing file. You can select the records to be sorted with both relational (greater than, less than, equal to or alphanumeic match) and logical comparisons. In this way you can create new *RADEX-10* files containing subsets of the main database. A high speed machine language subroutine is also used in this module.

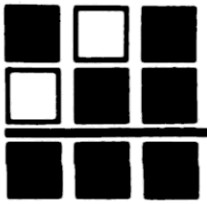
C.A.S. Database Editor

If you need to add, delete or change any fields in your *RADEX-10* database, without re-entering all the existing data, then this program is the answer. With it you can completely restructure or edit your database with ease, make fields longer or shorter, change the sequence of information, or even insert completely new fields.

All the modules work with either *RADEX-10* or the *Universal Database Manager*, but require 48K and *NEWDOS*.

Complete with extensive documentation, the modules are available to registered *RADEX-10* or *Universal Database Manager* owners for $99.00 each.

ASK FOR A DEMONSTRATION AT YOUR LOCAL DEALER



IJG COMPUTER SERVICES

569 N. MOUNTAIN • SUITE B • UPLAND, CA 91786 • (714) 982-7829

Reader Service—see page 226

A peripheral worth considering.

The Light Pen

Hugo T. Jackson
#401-1873 Nelson St.
Vancouver, B.C.
V6G 1M9

3G Light Pen
3G Company
Gaston, OR
$34.95

Perhaps the following scenario sounds familiar? Excited by the prospect of using your recently purchased computer for a business application, you diligently apply yourself to the task of writing your first program. Although it's hard work, you stick with it, emerging months later with the completed program in hand. You're eager to show your business partner or employee how to use and appreciate the sum result of your newfound knowledge.

You grow increasingly frustrated, however, when his trial attempt at the keyboard results in input errors, causing the program to bomb unmercifully. You quickly realize that if he is going to have any success with the computer, he will have to know as much about the machine as you do. That's an expensive proposition, even if it's only your time that's spent training him.

Your first home application isn't that successful either, if, like myself, you had intended to develop some educational programs for a child of six. Have you ever tried to explain "SYNTAX ERROR IN 140" to a child who can't even read?

The Light Pen

Fortunately, there is an answer to the problem, and it is called a light pen. What this marvelous device does is specify locations on the video monitor simply by placing the pen point directly over them.

The pen is quite simple, and, although designers may employ different refinements, they are all built on the same basic principle.

In the tip of the light pen is a photocell, phototransistor or similar electronic device. These components are all light sensitive. The pen's circuitry differentiates between two levels of light.

In computer applications, this translates to the pen's determination of whether or not a particular screen location is illuminated (by a letter or graphics character) or blank.

Aware of a light pen's potential, I reacted quickly to an advertisement from the 3G Company. After three and one-half weeks—interrogating the postman daily—the light pen was finally delivered. Opening the well-packaged container, I

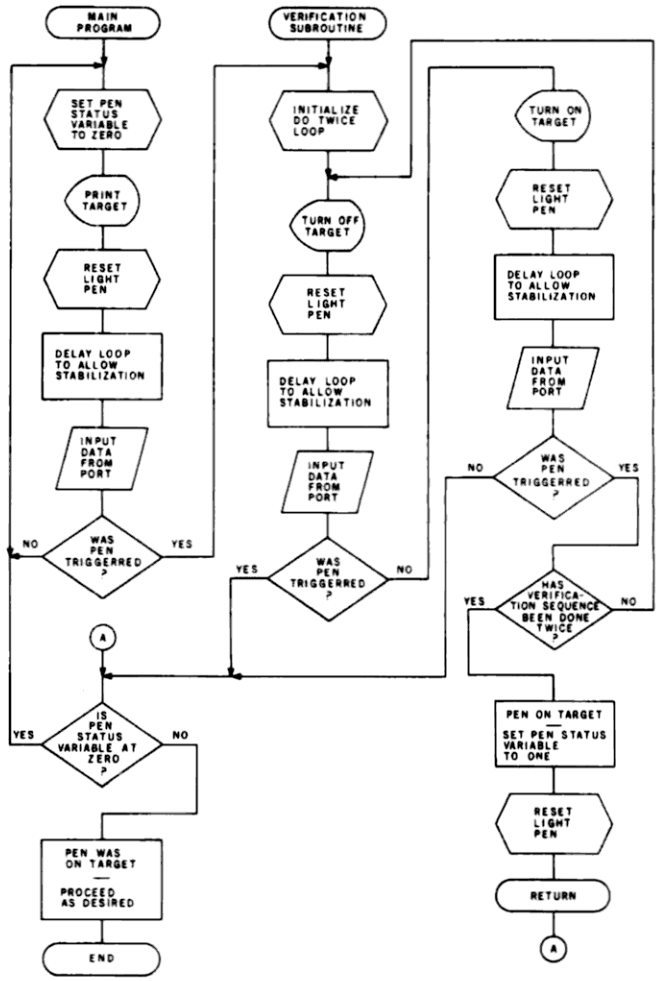


Fig. 1. Flowchart of main program requirements before branching to subroutine and logic of the verification subroutine.

Program Listing 1. Light Pen Detection Subroutine.

```
100 '*************************
110 'FIGURE 2:
120 '   LIGHT PEN DETECTION
130 '        SUBROUTINE
140 '
150 '     BY HUGO T. JACKSON
160 '*************************
170 '
180 'ASSIGNMENT OF VARIABLES:
190 '
200 'A = PEN STATUS VARIABLE
210 '    A=1 : SUBROUTINE
               DETERMINED PEN
               WAS INDEED ON
               TARGET.
220 '    A=2 : SUBROUTINE
               DETERMINED PEN
               WAS NOT
               ON TARGET
240 'B = FOR/NEXT LOOP THAT
         REPEATS DETECTION
         ROUTINE TWICE TO
         INSURE ACCURACY.
260 'C = FOR/NEXT LOOPS USED
         TO ALLOW THE DISPLAY
         TO STABILIZE BEFORE
         TESTING VALUE AT
         LIGHT PEN'S PORT
         ADDRESS.
280 'LO$ = A TWO CHARACTER
           GRAPHICS STRING USED
           AS THE TARGET.
300 'OF$ = TWO BACKSPACES TO
           ERASE THE TARGET TO
           CHECK IF THERE IS
           NOW LOW LOGIC AT
           THE PEN'S PORT
           ADDRESS.
310 '
320 '*************************
330 '
340 'MAIN PROGRAM:
350 'LINE NUMBERS 100-1000
    REPRESENT STATEMENT THAT
    MUST BE INCLUDED PRIOR
    TO TESTING THE PORT
    ADDRESS. THE CONDITIONAL
    BRANCH IS ALSO INCLUDED.
360 '
365 '*************************
370 CLS
380 'INITIALIZE THE TWO STRINGS
390 LO$=STRING$(2,143)
400 OF$=STRING$(2,8)
410 'SET PEN STATUS VARIABLE
    TO ZERO
420 A=0
430 'RESET LIGHT PEN
440 OUT99,0
450 PRINT@ 0, LO$;
460 'PRINT TARGET
470 FOR C=1 TO 5:NEXT C
480 'IF PEN IS TRIGGERED, GO
    TO DETECTION SUBROUTINE
    AND SEE IF IT WAS ON THE
    TARGET.
490 IF INP(99)>128 GOSUB 630
500 'IF PEN WAS ON TARGET
    SUBROUTINE WILL RETURN
    WITH VALUE OF A=1. THAT
    BEING THE CASE, THE
    PROGRAM WILL THEN BRANCH
    TO LINE *****.
510 IF A=1 GOTO 540
520 'PEN WAS NOT ON TARGET SO
    GO BACK AND LOOP UNTIL
    IT IS.
530 GOTO 420
540 PRINT@ 64,"LIGHT PEN ON TARGET"
550 END
560 '
570 '*************************
580 '
590 'DETECTION SUBROUTINE
600 '
610 '*************************
620 'FIRST SET UP FOR NEXT
    LOOP SO THAT WHOLE ROUTINE
    IS DONE TWICE.
630 FOR B=1 TO 2
640 'TURN OFF TARGET
650 PRINT OF$;
660 'RESET LIGHT PEN
670 OUT 99,0
680 'ALLOW DISPLAY TO STABILIZE
690 FOR C=1 TO 5:NEXT C
700 'TEST PORT:
    IF PEN STILL AT HIGH LOGIC
    THEN PEN NOT ON TARGET SO
    RETURN - ELSE CONTINUE
    DETECTION ROUTINE.
710 IF INP(99)>127 THEN RETURN
720 'NOW TURN TARGET BACK ON
730 PRINT LO$;
740 'RESET LIGHT PEN AGAIN
750 OUT 99,0
760 'ALLOW SCREEN TO STABILIZE
770 FOR C=1 TO 5:NEXT C
780 'CHECK PORT VALUE:
    IF PEN NOT AT HIGH LEVEL
    LOGIC THEN PEN IS NOT ON
    TARGET SO RETURN - ELSE
    CONTINUE.
790 IF INP(99)<=127 THEN RETURN
800 'REPEAT DETECTION ROUTINE
810 NEXT B
820 'DETECTION ROUTINE HAS
    BEEN PERFORMED TWICE AND
    PROGRAM CONTROL HAS NOT
    BEEN RETURNED TO MAIN
    PROGRAM. THEREFORE PEN
    MUST BE ON TARGET. SO
    FIRST SET PEN STATUS
    VARIABLE TO A=1.
830 A=1
840 'THEN RESET LIGHT PEN.
850 OUT 99,0
860 'FINALLY RETURN CONTROL
    TO MAIN PROGRAM.
870 RETURN
880 '
890 '
900 '*************************
```

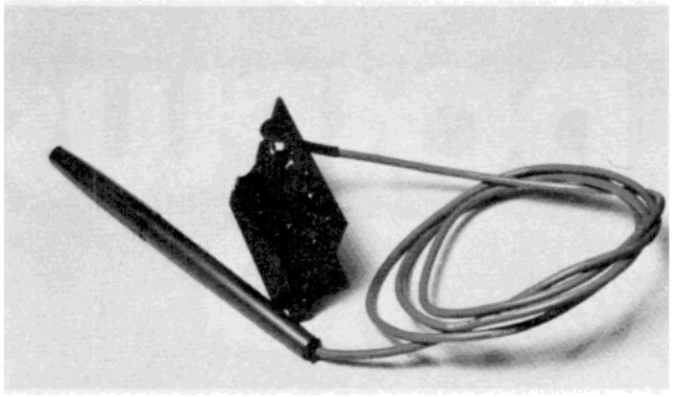


Photo 1. The 3G light pen. The connector at the end plugs into the expansion slot at the rear of the TRS-80.

found a demonstration program on cassette, two sheets of instructions and the light pen.

Wonder of wonders, the program loaded the first time, and when I ran it, the familiar tic-tac-toe grid appeared on the screen.

Although the pen's instructions indicated that my monitor might require adjustment of the contrast and brightness controls, I was happy to discover that the levels at which I usually have the monitor set were quite acceptable.

While tic-tac-toe is popular with my son, it is not the most intriguing game that I have ever played, so it wasn't long before I began thinking of other applications for the light pen. To ensure success, I first had to understand exactly how the pen worked.

Polling the Port

The light pen is assigned a port address of 99 and is controlled in BASIC programs with the INP and OUT commands. Using either equivalence (A = INP(99)) or conditional (IF INP(99)>128 GOTO 1310) statements, the program polls the port address of the pen and returns with either a value of 127 or 255. This indicates whether the light pen is on an unlit or illuminated portion of the screen.

However, before the port is polled, it is necessary to reset the pen logic, as the circuitry latches and holds high state logic; i.e., once the pen has been triggered, it returns a value of 255, whether or not it has been subsequently moved to an unlit portion of the screen. The statement OUT 99 resets the pen and must be executed prior to polling of the port address.

An additional problem is that the light pen doesn't care what type of light falls on it. As a result, it goes to a high logic state whether it is pointed at the video monitor, a desk lamp or even out the window on a sunny day. We work around this by providing "targets" on the monitor which indicate to the user where the pen must be placed, in order for his response to be recorded.

If you look at the flowchart in Fig. 1, you see that by turning on and off the "target" and comparing the screen condition with the value received from the light pen port, we are able to ascertain whether or not the pen is pointed at a particular target.

The flowchart leads to the development of subroutines such as the one in Program Listing 1. Prior to branching to the subroutine, the pen is reset and the port polled. If the pen is at high logic (255), it may well be over the target, so we branch to the subroutine to verify this. If the pen returns a value of less than 128, it obviously cannot be over a lit portion of the screen, so we loop back and continue polling the port until the pen is triggered again.

If, however, the pen does return with a value of 255, then we proceed with the subroutine's verification. To ensure that the pen is over the target we turn the target on again, reset the pen and poll the port. If the pen is

Program Listing 2. Speedo

```
1000 REM  ******************************************************
1010 REM
1020 REM                    THE GAME OF SPEEDO
1030 REM
1040 REM                     BY HUGO JACKSON
1050 REM
1060 REM  ******************************************************

1070 REM
1080 REM          ASSIGNMENT OF VARIABLES:
1090 REM
1100 REM          A1 = FOR/NEXT LOOP
1110 REM          A2 = FOR/NEXT LOOP
1120 REM          A4 = ROW VALUE OF CURRENT TARGET POSITION
1130 REM          A5 = COLUMN VALUE OF CURRENT TARGET POSITION
1140 REM          A6 = PIXEL VALUE FOR DISPLAY ARRAY:
1150 REM                          A6(0)=131 - TARGET AT
     TOP
1160 REM                          A6(1)=140 - TARGET AT
     MIDDLE
1170 REM                          A6(2)=176 - TARGET AT
     BOTTOM
1180 REM          A7 = ON POSITION INDICATOR
1190 REM          A8 = RANDOM DIRECTION INDICATOR
1200 REM          A9 = CURRENT TIMING VALUE
1210 REM          AA = CURRENT MATCH NUMBER
1220 REM          AB = CURRENT MATCH SCORE
1230 REM          AC = CURRENT TOTAL SCORE
1240 REM          AD = PRINT POSITION FOR CURRENT MATCH
     SCORE
1250 REM
1260 REM          B1 = CURRENT SCREEN LOCATION OF TARGET
1270 REM          B2 = NEW SCREEN LOCATION OF TARGET
1280 REM
1290 REM  ******************************************************
```

Program continues

```
1300 REM
1310 REM  INITIALIZE AND DEFINE VARIABLES ETC.
1320 REM
1330 RANDOM
1340 CLS
1350 DEFINT A-C
1360 DIM A6(3)
1370 A6(0)=131
1380 A6(1)=140
1390 A6(2)=176
1400 A9=0
1410 AA=1
1420 AB=0
1430 AD=571
1440 REM  POKE HORIZONTAL BORDER INTO DISPLAY MEMORY
1450 FOR A1=15360 TO 15487
1460 POKE A1,131
1470 POKE A1+960,131
1480 NEXT A1
1490 REM  POKE VERTICAL BORDER INTO DISPLAY MEMORY
1500 FOR A1=15360 TO 16256 STEP 64
1510 POKE A1,191
1520 POKE A1+47,191
1530 NEXT A1
1540 REM  PRINT TITLE AND OTHER PERMANENT DISPLAY MATERIAL
1550 PRINT@ 50,STRING$(14,61);
1560 PRINT@ 118,"SPEEDO";
1570 PRINT@ 178,STRING$(14,61);
1580 PRINT@ 434,"MATCH SCORES";
1590 PRINT@ 498,STRING$(14,45);
1600 PRINT@ 882,STRING$(14,45);
1610 REM  ENTRY POINT FOR EACH NEW MATCH
1620 PRINT@ 306,"TIME:";
1630 PRINT@ AD-9,"MATCH";AA;"=";AB;
1640 REM  DISPLAY TIME DELAY FOR THE BEGINNING OF EACH
     VOLLEY
1650 PRINT@ 399,"COUNTDOWN TO EVENT";
1660 FOR A1=10 TO 0 STEP -1
1670 FOR A2=1 TO 100:NEXT A2
1680 PRINT@ 471,A1;
1690 NEXT A1
1700 REM  ENSURE ALL PRIOR TEXT ERASED FROM DISPLAY
1710 PRINT@ 333,STRING$(28,32);
1720 PRINT@ 399,STRING$(18,32);
1730 PRINT@ 470,STRING$(4,32);
1740 REM  RANDOMLY DETERMINE INITIAL X-Y CO-ORDINATES
     OF TARGET
1750 A4=RND(14)
1760 A5=RND(43+1)
1770 B1=15360+(64*A4)+A5
1780 A7=RND(3)-1
1790 REM  PRINT TARGET
1800 POKE B1,A6(A7)
1810 POKE B1+1,A6(A7)
1820 REM  RESET LIGHT PEN
1830 OUT 99,0
1840 REM  TIMING LOOP TO ALLOW DISPLAY/PEN TO STABILIZE
1850 FOR A1=1 TO 8:NEXT A1
1860 REM  TEST FOR TRIGGERING - IF NO THEN BRANCH
1870 IF INP(99)<128 GOTO 2330
1880 REM  ROUTINE TO TEST IF PEN WAS DIRECTLY OVER TARGET
1890 FOR A1=1 TO 2
1900 REM  LOAD CURRENT TARGET POSITION WITH BLANKS
1910 POKE B1,32
1920 POKE B1+1,32
1930 REM  TIMING LOOP TO ALLOW DISPLAY TO STABLIZE
1940 FOR A2=1 TO 15:NEXT A2
1950 REM  RESET LIGHT PEN
1960 OUT 99,0
1970 REM  TIMING LOOP TO ALLOW DISPLAY/PEN TO STABILIZE
1980 FOR A2=1 TO 5:NEXT A2
1990 REM  TEST FOR TRIGGERING - IF TRIGGERED THEN PEN
     CANNOT
        BE OVER TARGET AS IT IS NOW OFF - SO BRANCH TO
        PENALTY SECTION
2000 IF INP(99)>127 GOTO 2240
2010 REM  TURN TARGET BACK ON
2020 POKE B1,A6(A7)
2030 POKE B1+1,A6(A7)
2040 REM  TIMING LOOP TO STABILIZE DISPLAY/PEN
2050 FOR A2=1 TO 10:NEXT A2
2060 REM  TEST FOR TRIGGERING - IF PEN NOT TRIGGERED THEN
        IT IS NOT OVER TARGET AS IT IS NOW ON - BRANCH
     TO
        PENALTY SECTION
2070 IF INP(99)<=127 GOTO 2240
2080 REM  REPEAT TEST SECTION TO INSURE ACCURACY
2090 NEXT A1
2100 REM  RESET LIGHT PEN
2110 OUT 99,0
2120 REM  INCREMENT MATCH POINT VALUE
2130 AB=AB+1
2140 REM  PRINT CURRENT SCORE
2150 PRINT@ AD,AB;
2160 REM  ERASE TARGET AT CURRENT POSITION
2170 POKE B1,32
2180 POKE B1+1,32
2190 REM  NOTIFY USER OF SUCCESSFUL POINT
2200 PRINT@ 339,"YOUR POINT";
2210 REM  BEGIN NEW VOLLEY
2220 GOTO 1650
2230 REM  THIS IS THE PENALTY SECTION WHICH CONSISTS OF
        ADDING TEN TIME UNITS TO THE CURRENT VALUE
        AND RETURNING TO THE MATCH LOOP
2240 A9=A9+10
2250 IF A9>=101 GOTO 2600
2260 POKE B1,32
2270 POKE B1+1,32
2280 PRINT@ 333,"PENALTY FOR INACCURACY";
2290 PRINT@ 312,A9;
2300 FOR A2=1 TO 100:NEXT A2
2310 GOTO 1650
2320 REM  PROGRAM SECTION TO INCREASE TIME VALUE AND
```

Program continues

```
        CALCULATE NEW POSITION FOR TARGET
2330 A9=A9+1
2340 IF A9>=101 GOTO 2600
2350 PRINT@ 312,A9;
2360 REM  RANDOMLY CALCULATE DIRECTION
2370 A8=RND(9)
2380 REM  AND CALCULATE NEW POSITION ACCORDING TO RANDOM
     CHOICE
2390 IF A8=9 THEN A4=20:A5=50
2400 IF A8=1 THEN A7=A7-1:A5=A5-1
2410 IF A8=2 THEN A5=A5-1
2420 IF A8=3 THEN A7=A7+1:A5=A5-1
2430 IF A8=4 THEN A7=A7+1
2440 IF A8=5 THEN A7=A7+1:A5=A5+1
2450 IF A8=6 THEN A5=A5+1
2460 IF A8=7 THEN A7=A7-1:A5=A5+1
2470 IF A8=8 THEN A7=A7-1
2480 IF A7=-1 THEN A7=2:A4=A4-1
2490 IF A7=3 THEN A7=0:A4=A4+1
2500 REM  MAKE SURE TARGET IS WITHIN BOARD BOUNDARIES
2510 IF A4>13 OR A4<1 THEN A4=RND(14)
2520 IF A5>44 OR A5<2 THEN A5=RND(43)+1
2530 B2=15360+(64*A4)+A5
2540 REM  DISPLAY TARGET AT NEW POSITION
2550 POKE B1,32
2560 POKE B1+1,32
2570 B1=B2
2580 GOTO 1800
2590 REM  END OF MATCH SO INCREMENT MATCH VARIABLE AND
        INITIALIZE MATCH SCORE VARIABLE ETC.
2600 AC=AC+A8
2610 AA=AA+1
2620 IF AA=6 GOTO 2750
2630 A9=0
2640 PRINT@ 312,"0   ";
2650 A8=0
2660 AD=AD+64
2670 REM  NOTIFY USER OF NEW MATCH
2680 PRINT@ 339,"NEW MATCH";
2690 POKE B1,32
2700 POKE B1+1,32
2710 FOR A2=1 TO 100:NEXT A2
2720 REM  RETURN TO MATCH LOOP
2730 GOTO 1630
2740 REM  END OF GAME ROUTINE
2750 POKE B1,32
2760 POKE B1+1,32
2770 PRINT@ 267,STRING$(26,61);
2780 REM  NOTIFY USER OF END AND PRINT TOTAL AND AVERAGE
     SCORE
2790 PRINT@ 340,"GAME END";
2800 PRINT@ 395,STRING$(26,61);
2810 PRINT@ 524,"AVERAGE MATCH SCORE:";AC/5;
2820 PRINT@ 590,"TOTAL GAME SCORE";AC;
2830 PRINT@ 715,STRING$(26,61);
2840 REM  RESET LIGHT PEN
2850 OUT 99,0
2860 REM  CHECK FOR TRIGGERING OF ANY KIND - IF NONE RESET
        LIGHT PEN AND CHECK AGAIN
2870 IF INP(99)<=127 GOTO 2850
2880 REM  TIMING LOOP TO ALLOW FOR USER TO MOVE PEN AWAY'
        FROM THE LIGHT SOURCE
2890 FOR A2=1 TO 500:NEXT A2
2900 REM  RESET LIGHT PEN
2910 OUT 99,0
2920 REM  IF LIGHT PEN IS NO LONGER TRIGGERED THEN RETURN
     TO
        BEGINNING OF THIS LOOP - OTHERWISE USER DESIRES
        ANOTHER GAME SO AFTER CLEAR - GOTO PROGRAM
        BEGINNING
2930 IF INP(99)<=127 GOTO 2850
2940 CLEAR
2950 GOTO 1340
```

Program continues

still on target, the port returns a value of 255. If it doesn't, we return to the main program, having been unsuccessful in verifying the "hit."

To make absolutely certain the pen is on target, it is best to repeat this whole sequence once again, which is why the detection portion of the subroutine is nested in a "do twice" FOR/NEXT loop. If the pen falls through the FOR/NEXT loop without having been returned to the main program, we can safely assume that the pen is on target.

To record positive verification, the pen status variable is set to one, the light pen is reset, and program control is returned to the main program. How the program proceeds after verification of the pen being on target is up to the programmer.

The final software consideration is to provide for situations where more than one target is presented to the user. This is accomplished in the main program with a FOR/NEXT loop which sequentially turns on each target and then checks the port address for triggering.

If the pen has been triggered, the program branches to the detection subroutine and determines if the pen is over the target currently being printed. Even though the pen may be placed on a target other than the one presently being polled, the FOR/NEXT loop quickly cycles through all the target locations until the subroutine returns with positive verification of the pen at the current target position.

Two Programs

To date, my experimentation with the light pen has resulted in two programs. The first is a short general history quiz which also instructs you in the use of the light pen. The second program is a challenging target game. None of the remarks are used for branching purposes in either program, so they may be deleted when you enter the program. If you leave them in the target game, they are a handicap as they slow down the program's execution substantially.

Apparently, other manufacturers besides 3G offer light pens that utilize the cassette port for polling and resetting. If you own one of these other pens, these two programs will run properly, if you change all I/O commands to OUT (255) and INP(255). The values returned from the port address on polling are undoubtedly 0 and 1. These values should be substituted for the 127 and 128 values used in my programs.

For those of you without a light pen, Compututor History Quiz can easily be adapted for regular keyboard input. The questions are multiple choice.

The Game of Speedo

The target in Speedo moves

randomly within the confines of a graphics box. The object of the game is to "hit" it as many times as possible within the time limit. The game is comprised of five matches, each having a time limit of one hundred units. Watch out! You not only have to hit the target, but you have to remain there until the subroutine ensures that you are on target. Any time the pen is triggered and the subroutine finds that you are not on target, you will be penalized ten time units for your inaccuracy.

An interesting departure from the standard detection method is given in lines 2820 and higher. Here time is used as the sole determining factor, since the pen need only be placed on a lit portion of the screen and held there for about five seconds. If the pen is still triggered after the timing loop in 2870 has been executed, the program clears all variables and restarts the game.

Through my experience, I have come to the conclusion that a light pen is probably the most valuable peripheral that you can add to your current system. It is possible to eliminate all keyboard input except for RUN.■

Program Listing 3. Computer History Quiz

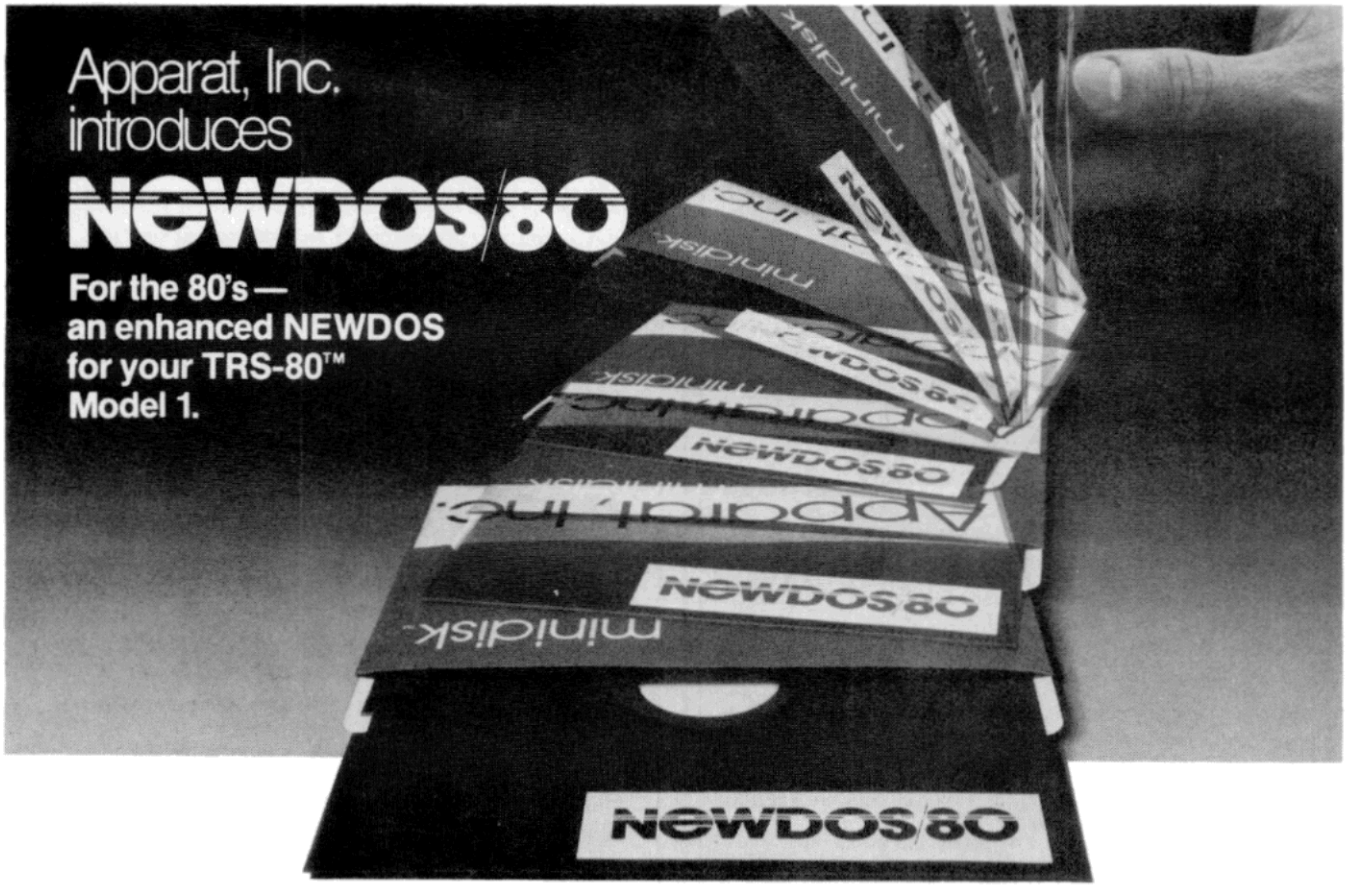
```
100 REM  ************************************************
110 REM
120 REM                      COMPUTUTOR
130 REM                    HISTORY   QUIZ
140 REM
150 REM                  BY HUGO T. JACKSON
160 REM
170 REM  ************************************************
180 REM
190 REM  ASSIGNMENT OF VARIABLES:
200 REM         A1 = CURRENT QUESTION NUMBER
210 REM         A2 = ARRAY TO TRACK QUESTIONS ASKED AND
                     ANSWERS USED
220 REM         A3 = QUESTION FORMAT INDICATOR:
                        A3=0  DATE TOLD - EVENT FOR ANSWER
                        A3=1  EVENT TOLD - DATE FOR ANSWER
230 REM         A4 = RANDOM SELECTION OF QUESTION ASKED
240 REM         A5 = FOR/NEXT LOOP
250 REM         A6 = FOR/NEXT LOOP
260 REM         A7 = PRINT POSITION
270 REM         A8 = CORRECT ANSWER
280 REM         A9 = NUMBER OF QUESTIONS CORRECTLY ANSWERED
290 REM         AB = FOR/NEXT LOOP
300 REM         AC = FOR/NEXT LOOP
310 REM         AD = FOR/NEXT LOOP
320 REM
330 REM         B1 = ANSWER ARRAY
340 REM         B2 = CORRECT ANSWER
350 REM         B3 = CORRECT QUESTION
360 REM
370 REM         C1 = TEST PERCENTAGE
380 REM
390 REM         LO = LIGHT PEN TARGET
400 REM         LF = STRING TO TURN OFF TARGET
410 REM
420 REM         P1 = LIGHT PEN STATUS INDICATOR
430 REM
440 REM  ************************************************
1000 REM  INITIALIZE & DEFINE VARIABLES ETC.
1010 CLS
1020 CLEAR 1000
1030 DEFSTR B,L
1040 DEFSNG C
1050 DEFINT A
1060 DIM B1(4)
1070 DIM A2(50)
1080 RANDOM
1090 REM  DEFINE LIGHT PEN TARGET
1100 LN=STRING$(2,143)
1110 REM  DEFINE BACKSPACE TO ERASE LIGHT PEN TARGET
1120 LF=STRING$(2,8)
1130 REM  PRINT FIRST MESSAGE
1140 PRINT@ 21,STRING$(21,45)
1150 PRINT@ 85,"WELCOME TO COMPUTUTOR";
1160 PRINT@ 149,STRING$(21,45)
1170 PRINT@ 261,"THIS IS A TEST OF YOUR GENERAL KNOWLEDGE
     OF HISTORICAL"
1180 PRINT@ 330,"EVENTS. YOU WILL BE ANSWERING MULTIPLE
     CHOICE";
1190 PRINT@ 394,"QUESTIONS AND INDICATING YOUR CHOICE
     BY USING";
1200 PRINT@ 460,"THE LIGHT PEN. TO USE A LIGHT PEN SIMPLY";

1210 PRINT@ 518,"PLACE IT ON THE VIDEO SCREEN OVER THE
     LIGHT SQUARE.";
1220 PRINT@ 648,"TRY IT NOW. SIMPLY PLACE THE LIGHT PEN '
     OVER THE";
1230 PRINT@ 712,"SQUARE BELOW. WHEN YOU HAVE DONE THAT,
      I'LL KNOW";
1240 PRINT@ 776,"THAT YOU ARE READY WILLING AND ABLE TO '
     CONTINUE.";
1250 REM  RESET LIGHT PEN
1260 OUT99,0
1270 REM  PRINT LIGHT PEN TARGET
1280 PRINT@ 927,LN;
1290 REM  WAS PEN TRIGGERED? - THEN BRANCH
1300 IF INP(99)>127 GOSUB 5020
1310 REM  P1=0 IF LIGHT PEN WAS NOT OVER TARGET
1320 IF P1=0 GOTO 1260
1330 REM  PEN ON TARGET SO PRINT NEW TEXT
1340 CLS
1350 PRINT@ 27,"VERY GOOD!"
1360 PRINT@ 137,"LET'S TRY A SAMPLE QUESTION TO MAKE SURE
     WE";
1370 PRINT@ 199,"UNDERSTAND EACH OTHER. I'LL ASK YOU"A
     QUESTION";
1380 PRINT@ 266,"AND SHOW YOU FIVE POSSIBLE ANSWERS ONLY
     ONE";
1390 PRINT@ 326,"OF WHICH IS CORRECT. AFTER YOU HAVE DECIDED
     WHAT THE";
1400 PRINT@ 391,"CORRECT ANSWER IS JUST POINT THE LIGHT '
     PEN AT THE";
1410 PRINT@ 460,"SQUARE OF LIGHT BESIDE THE RIGHT ANSWER.";

1420 PRINT@ 587,"LET ME KNOW WHEN YOU ARE READY TO CONTINUE";

1430 PRINT@ 916,"READY TO CONTINUE? ";CHR$(94);
1440 REM  RESET LIGHT PEN
1450 OUT99,0
1460 REM  RESET LIGHT PEN STATUS VARIABLE
1470 P1=0
1480 REM  PRINT TARGET
1490 PRINT@ 937,LN;
1500 REM  TEST TO SEE IF LIGHT PEN TRIGGERED
1510 IF INP(99)>127 GOSUB 5020
1520 REM  P1=1 IF LIGHT PEN WAS OVER TARGET
1530 IF P1=1 GOTO 1560
1540 REM  AS IT WAS NOT, LOOP UNTIL IT IS
1550 GOTO 1490
1560 CLS
1570 PRINT@ 320,"WHAT COLOUR IS THE SKY?";
1580 PRINT@ 458,"BROWN";
1590 PRINT@ 522,"BLUE";
1600 PRINT@ 586,"ORANGE WITH BLUE POLKA DOTS";
1610 PRINT@ 650,"IT HAS NO COLOUR";
1620 PRINT@ 714,"WHAT DOES IT MATTER";
1630 REM  RESET LIGHT PEN
1640 OUT 99,0
1650 REM  RESET LIGHT PEN STATUS VARIABLE
1660 P1=0
1670 REM  PRINT 5 TARGETS, ONE FOR EACH POSSIBLE ANSWER
1680 FOR A5=448 TO 704 STEP 64
1690 PRINT@ A5,LN;
1700 REM  TIMING LOOP TO ALLOW DISPLAY TO STABLIZE
1710 FOR AB=1 TO 9:NEXT AB
1720 REM  TEST FOR TRIGGERING, BRANCH IF SO
1730 IF INP(99)>127 THEN GOSUB 5020
1740 REM  P1=1 IF LIGHT PEN OVER ONE OF THE TARGETS -
     A5 WILL
          BE EQUAL TO THE VALUE OF THE PRINT POSITION WHERE
          TRIGGERING TARGET IS LOCATED
```

Program continues



Apparat, Inc., announces the most powerful Disk Operating System for the TRS-80®. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2.1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2.1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2.1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:

- New BASIC commands that supports files with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disks drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOS READY" or ">READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST".
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful chaining commands.
- Print Spooler.
- DFG function; simultaneous striking of the D, F and G keys will allow the user to enter a mini-DOS to perform some DOS commands without disturbing the resident program. (e.g. dir while in scripsit.)
- Upward compatible with NEWDOS 2.1 and TRSDOS 2.3.
- Includes machine language Superzap/80 and all Apparat 2.1 utilities.
- Enter debug any time by pressing 123 keys. Also allows disk I/O.
- Diskette "Purge" command.
- Specifiable system options (limited sysgen type commands).
- Increased directory capacity.
- Copy by file commands.

NEWDOS/80 with all of the NEWDOS + utility programs, many of which have been enhanced, is priced at just $149.00 and is available at most TRS-80 dealers.

As with 2.1, NEWDOS/80 relies on the TRSDOS and Disk Basic Reference Manual published by Radio Shack. NEWDOS/80 documentation supports its enhancements and upgrades only.

VISA master charge

✓264

Apparat, Inc.

MICROCOMPUTER TECHNOLOGY INCORPORATED

✓28

TO PURCHASE NEWDOS/80, COMPLETE AND MAIL TO:

Apparat, Inc.
4401 S. Tamarac Parkway
Denver, CO 80237
303/758-7275 303/741-1778

—OR—

Microcomputer Technology, Inc.
3304 W. MacArthur Blvd.
Santa Ana, CA 92704
714/979-9923

☐ Check ☐ Money Order ☐ Master Charge ☐ Visa

Card No. _____ Expiration Date _____

Colo. residents add 6.5% sales tax. Cal. residents add 6% sales tax.
Add $10.00 postage and handling.

Please rush _____ NEWDOS/80 @ $149 EACH TO:

Name _____

Address _____

City _____ State _____ Zip _____

Phone _____

0/1A

✓Reader Service—see page 226

```
1750 IF P1=1 THEN GOTO 1800
1760 REM  PRINT NEXT TARGET AND CHECK FOR TRIGGERING
1770 NEXT A5
1780 REM  NO TRIGGERING - REPEAT SEQUENCE
1790 GOTO 1680
1800 CLS
1810 REM  TARGET AT PRINT POSITION 512 IS ONLY CORRECT
     ONE SO
          PRINT APPROPRIATE MESSAGE ACCORDING TO USER RESPONSE
1820 IF A5=512 GOTO 1870
1830 PRINT@ 138,"EVEN THOUGH YOU ANSWERED THAT LAST QUESTION";

1840 PRINT@ 201,"INCORRECTLY I AM SURE YOU NOW KNOW HOW
     TO USE";
1850 PRINT@ 271,"THE LIGHT PEN  TO ANSWER QUESTIONS."
1860 GOTO 1900
1870 PRINT@ 155,"VERY GOOD!"
1880 PRINT@ 203,"AT LEAST I KNOW YOUR NOT GOING TO HAVE '
     ANY";
1890 PRINT@ 264,"PROBLEMS USING THE LIGHT PEN TO ANSWER
     QUESTIONS";
1900 PRINT@ 409,"SHALL WE BEGIN?";
1910 PRINT@ 917,"READY TO CONTINUE ";CHR$(94);
1920 REM  RESET LIGHT PEN
1930 OUT 99,0
1940 REM  RESET LIGHT PEN STATUS VARIABLE
1950 P1=0
1960 REM  PRINT TARGET
1970 PRINT@ 938,LN;
1980 REM  TEST FOR TRIGGERING
1990 IF INP(99)>127 GOSUB 5020
2000 REM  P1=1 IF PEN WAS OVER TARGET
2010 IF P1=1 GOTO 2050
2020 REM  AS IT WAS NOT - LOOP UNTIL IT IS
2030 GOTO 1970
2040 REM  AS IT WAS PRINT NEXT MESSAGE
2050 CLS
2060 PRINT@ 265,"WE CAN DO THIS TEST ONE OF TWO WAYS.
     EITHER I";
2070 PRINT@ 326,"CAN TELL YOU THE DATE AND YOU TELL ME
     WHAT HAPPENED";
2080 PRINT@ 392,"OR I'LL TELL YOU WHAT HAPPENED AND YOU
     PICK THE";
2090 PRINT@ 473,"CORRECT DATE.";
2100 PRINT@ 523,"IT DOESN'T MATTER TO ME WHICH WAY WE
     DO IT";
2110 PRINT@ 601,"SO YOU CHOOSE";
2120 PRINT@ 714,"I TELL YOU THE DATE -- YOU TELL ME WHAT
     HAPPENED";
2130 PRINT@ 778,"I TELL YOU WHAT HAPPENED -- YOU TELL
     ME THE DATE";
2140 REM  RESET LIGHT PEN
2150 OUT 99,0
2160 REM  RESET LIGHT PEN STATUS VARIABLE
2170 P1=0
2180 REM  PRINT TWO TARGETS - ONE FOR EACH POSSIBLE RESPONSE
2190 FOR AH=704 TO 768 STEP 64
2200 PRINT@ AH,LN;
2210 REM  TEST FOR TRIGGERING
2220 IF INP(99)>127 GOSUB 5020
2230 REM  P1=1 IF PEN ON TARGET
2240 IF P1=1 GOTO 2290
2250 REM  PRINT NEXT TARGET
2260 NEXT AH
2270 REM  NO TRIGGERING - REPEAT SEQUENCE'
2280 GOTO 2190
2290 CLS
2300 REM  CONVERT DISPLAY TO 32 CHARACTER FORMAT
2310 PRINTCHR$(23)
2320 REM  INFORM USER OF DELAY WHILE COMPUTER INITIALIZES
          ADDITIONAL VARIABLES ETC.
2330 PRINT@ 394,"STANDBY...";
2340 REM  DETERMINE FORMAT CHOICE OF USER BY PRINT POSITION
          OF THE TRIGGERING TARGET
2350 IF AH=768 THEN A3=1 ELSE A3=0
2360 REM  INITIALIZE QUESTION COUNTER'
2370 A1=1
2380 REM  RANDOMLY DETERMINE ORDER IN WHICH QUESTIONS
     ARE ASKED
2390 A4=RND(50)
2400 REM  DETERMINE IF SELECTION HAS BEEN PICKED BEFORE
2410 IF A2(A4)>0 GOTO 2390 ELSE A2(A4)=11
2420 RESTORE
2430 REM  GET APPROPRIATE STRINGS (QUESTION AND ANSWER) '
     FROM
          DATA STATEMENTS
2440 FOR A5=1 TO A4
2450 READ B8
2460 READ B9
2470 NEXT A5
2480 IF A3=1 THEN B3=B9 ELSE B3=B8
2490 IF A3=1 THEN B2=B8 ELSE B2=B9
2500 B1(0)=B2
2510 FOR A6=1 TO 4
2520 RESTORE
2530 REM  RANDOMLY DETERMINE OTHER POSSIBLE ANSWERS FOR '
     USER'S
          CHOICE
2540 A4=RND(50)
2550 REM  DETERMINE IF IT HAS BEEN PREVIOUSLY CHOSEN
2560 IF A2(A4)=1 OR A2(A4)=11 GOTO 2540 ELSE A2(A4)=A2(A4)+1
2570 FOR A5=1 TO A4
2580 READ B8
2590 READ B9
2600 IF A3=1 THEN B1(A6)=B8 ELSE B1(A6)=B9
2610 NEXT A5
2620 NEXT A6
2630 CLS
2640 PRINT@ 0,"QUESTION NO:";A1
2650 REM  BRANCH ACCORDING TO QUESTION FORMAT
2660 ON A3+1 GOTO 2680 ,2760
2670 REM  FOR 'DATE AS QUESTION' FORMAT, RANDOMLY DETERMINE
          HOW THE QUESTION IS PHRASED
```

Program continues

```
2680 ON RND(3) GOTO 2690 ,2710 ,2730
2690 PRINT@ 128,"WHAT IS ";B3;" MOST NOTED FOR?";
2700 GOTO 2790
2710 PRINT@ 128,"WHAT HAPPENED ON ";B3;"?";
2720 GOTO 2790
2730 PRINT@ 128,B3;" IS AN IMPORTANT DATE, WHAT HAPPENED?";

2740 GOTO2790
2750 REM  DUE TO PROBLEMS IN GRAMMER, 'OCCURENCE AS QUESTION'
       FORMAT IS PHRASED IN ONLY ONE WAY
2760 PRINT@ 128,B3;" - WHEN?";
2770 REM  THE ANSWERS FROM WHICH THE USER MAY CHOOSE ARE
     STORED
       IN ARRAY B1.  AS THE ORDER IN WHICH THEY ARE
     PRINTED
       OUT MIGHT INDICATE TO THE USER WHAT THE CORRECT
       ANSWER IS, THE ORDER IN WHICH THEY ARE PRINTED'
2780 REM  IS RANDOMLY DETERMINED.
2790 A7=264
2800 FOR A5=0 TO 4
2810 A6=RND(5)-1
2820 IF B1(A6)="" GOTO 2810
2830 PRINT@ A7,B1(A6);
2840 IF A6=0 THEN A8=A7+61
2850 B1(A6)=""
2860 A7=A7+64
2870 NEXT A5
2880 REM  RESET LIGHT PEN STATUS VARIABLE
2890 P1=0
2900 REM  RESET LIGHT PEN
2910 OUT 99,0
2920 A7=261
2930 REM  SET UP TARGET PRINTING LOOP FOR 5 DIFFERENT
     TARGETS
2940 FOR A5=0 TO 4
2950 REM  PRINT TARGET
2960 PRINT@ A7,LN;
2970 A7=A7+64
2980 REM  ALLOW DISPLAY TO STABLIZE
2990 FOR A6=1 TO 8
3000 REM  TEST FOR TRIGGERING
3010 IF INP(99)>127 GOSUB 5020
3020 REM  P1=1 IF LIGHT PEN WAS OVER TARGET PRESENTLY
     PRINTED
3030 IF P1=1 GOTO 3080
3040 REM  AS IT WASN'T PRINT NEXT TARGET
3050 NEXT A5
3060 REM  NOTHING TRIGGERED, REPEAT SEQUENCE UNTIL TRIGGERED
3070 GOTO 2920
3080 IF A8=A7 THEN A9=A9+1 ELSE AA=AA+1
3090 REM  DETERMINE IF USER INPUT WAS CORRECT
3100 IF A8=A7 GOTO 3130
3110 GOTO 3250
3120 REM  AS THE ANSWER WAS CORRECT RANDOMLY DETERMINE
       CONGRATULATORY REMARK
3130 ON RND(5) GOTO 3140 ,3160 ,3180 ,3200 ,3220
3140 PRINT@ 709,"THAT'S RIGHT";
3150 GOTO 3390
3160 PRINT@ 709,"CORRECT";
3170 GOTO 3390
3180 PRINT@ 709,"VERY GOOD!";
3190 GOTO 3390
3200 PRINT@ 709,"YOU'RE ABSOLUTELY RIGHT";
3210 GOTO 3390
3220 PRINT@ 709,"GOOD WORK";
3230 GOTO 3390
3240 REM  AS IT WAS INCORRECT - NOTIFY USER AND PRINT
     CORRECT
       ANSWER
3250 ON RND(5) GOTO 3260 ,3280 ,3300 ,3320 ,3340
3260 PRINT@ 709,"SORRY, WRONG ANSWER";
3270 GOTO 3350
3280 PRINT@ 709,"THAT'S NOT RIGHT";
3290 GOTO 3350
3300 PRINT@ 709,"UNFORTUNATELY THAT'S THE WRONG ANSWER";'

3310 GOTO 3350
3320 PRINT@ 709,"NO, THAT IS NOT RIGHT";
3330 GOTO 3350
3340 PRINT@ 709,"INCORRECT";
3350 PRINT@ 773,"THE CORRECT ANSWER WAS:";
3360 PRINT@ 840,CHR$(34);B2;CHR$(34)
3370 GOTO 3390
3380 REM  CLEAR ARRAY OF INCORRECT ANSWERS USED IN LAST '

       QUESTION AS THEY MAY BE USED AGAIN IN NEXT
       QUESTION
3390 FOR A5=0 TO 50
3400 IF A2(A5)=11 OR A2(A5)=1 THEN A2(A5)=A2(A5)-1
3410 NEXT A5
3420 REM  INCREMENT QUESTION COUNTER
3430 A1=A1+1
3440 REM  HAS THE END BEEN REACHED? IF SO GOTO TO END
     PORTION
       OF PROGRAM
3450 IF A1=51 GOTO 3480
3460 GOTO 2390
3470 REM  TIMING LOOP TO PRESERVE DISPLAY
3480 FOR A6=1TO500:NEXT A6
3490 REM  PRINT FINAL MESSAGE AND DISPLAY SCORE AND PERCENTAGE
3500 CLS
3510 PRINT@ 193,"WELL, THAT'S IT. YOU HAVE ANSWERED ALL";

3520 PRINT@ 257,"ALL FIFTY QUESTIONS.  THIS TIME YOU GOT";

3530 PRINT@ 320,A9;"RIGHT AND";AA;"WRONG WHICH WORKS OUT
     TO";
3540 PRINT@ 384,A9*2;"%";
3550 C1=A9*2
3560 REM  PRINT ADDITIONAL COMMENT AS WARRANTED BY SCORE'
3570 IF C1<50 THEN B2="BETTER LUCK NEXT TIME"
3580 IF C1>60 THEN B2="GOOD WORK"
3590 IF C1>70 THEN B2="VERY GOOD WORK"
3600 IF C1>80 THEN B2="EXCELLENT"
```

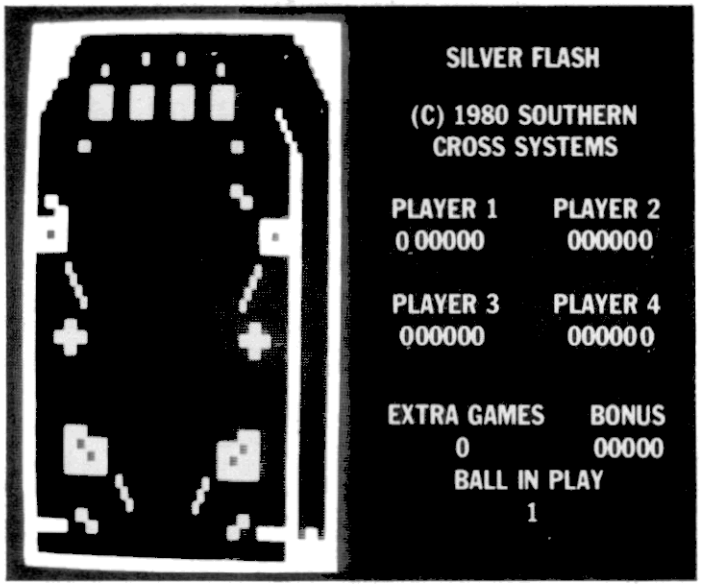
Program continues

✓Reader Service—see page 226

SILVER FLASH

THE FASTEST PINBALL GAME IN TOWN!

by Sheila Finnerty!



More features - More challenging

- ENTIRELY in machine language for fast action
- 1-4 players
- Upper and lower flippers
- Resetting flip flops and rollovers
- Player controls ball direction
- Extra balls to be won
- Extra games to be won
- Tilt function
- Double Bonus on third ball
- Blinking 'Black Holes'
- Hitting ball with flippers increases speed
- Ultimate sound feature
- 1/6 scale of a real pinball game

For use on TRS 80 Level II 16K*
Dealer Inquiries invited

$19.95 For Cassettes

*TRS 80 is a trademark of Tandy Corp.

Send check or moneyorders to: **SOUTHERN CROSS SYSTEMS** ✓448
Sorry. No phone orders
Pacific Trade Center, Suite 301, 190 S. King Street.
Honolulu, Hawaii 96813 Ph 808-524-5282

Have computer, will travel.

Executive Computer System Carrying Cases.

- Makes your microcomputer truly portable.
- Protects your equipment: locking latches limit access.
- Rugged black vinyl with metal corners outside.
- Protective foam rubber, black velveteen covered, inside.
- Computer can be operated without removing from case.
- And cases are custom designed for full systems.

Apple* Executive Case holds:

- Apple microcomputer.
- 9" Sanyo monitor.
- 2 disk drives.
- Power strip.
- 2 boxes diskettes.
- Manuals.
- Dimensions: 28"x 21"x 10½".
- Weight: 17 pounds.
- Price: $179

TRS-80 Executive Case** holds:

- TRS-80 Microcomputer.
- Expansion interface.
- 2 disk drives.
- Power strip.
- 2 boxes diskettes.
- Manuals.
- Dimensions: 28"x 21½"x 8½".
- Weight: 17 pounds.
- Price: $179

Terms: FOB Los Angeles—Master Charge, Visa or check with order. Allow 3-4 weeks for delivery.

*Registered, Apple Computers, Inc.
**Registered Trademark, Tandy Corporation.

COMPUTER TEXTile ✓390

10960 Wilshire Blvd. Suite 1504
Los Angeles, CA 90024 (213) 477-2196

IN STOCK FOR IMMEDIATE SHIPMENT

TRS-80*

16K MEMORY KITS $49.95
4116's 6 MONTH WARRANTY INSTRUCTIONS INCLUDED

DISK DRIVES
PERCOM TFD-100 $325
PERCOM TFD-200 $595
CCI-100 $305
CCI-200 $495
2 DRIVE CABLE $24.95 4 DRIVE CABLE $34.95

PRINTERS $695
MICROTEK MT80P BI-DIRECTIONAL
125 CPS UPPER & LOWER CASE
1 YEAR WARRANTY
CABLE—$24.95

DISKETTES
MEMOREX OR BASF 10/$26.50
YOUR SATISFACTION GUARANTEED OR FULL REFUND

MICROCOMPUTER SERVICES CORPORATION
7314 MATTHEWS-MINT HILL RD. CHARLOTTE, NC 28212

✓442 *TRADEMARK TANDY/RADIO SHACK CORP. PERCOM TM PERCOM DATA CCI TM CPU IND.

704-545-0826

```
3610 IF C1=100 THEN B2="PERFECT -- NOT ONE WRONG"
3620 PRINT@ 576,B2
3630 PRINT
3640 PRINT
3650 END
5000 REM  SUBROUTINE TO INSURE LIGHT PEN IS OVER TARGET.
          ACCOMPLISHED BY TURNING OF THE LIGHT PEN - CHECKING
          FOR CHANGE IN PORT VALUE - TURNING IT
5010 REM  ON AGAIN TO SEE IF THE PEN IS STILL IN THE SAME

          POSITION AND REPEATING THE WHOLE SEQUENCE AGAIN
          JUST TO MAKE SURE.
5020 P1=0
5030 REM  LOOP TO PERFORM OPERATION TWICE
5040 FOR AB=1 TO 2
5050 REM  TURN LIGHT PEN OFF
5060 PRINT LF;
5070 REM  TIMING LOOP TO STABILIZE DISPLAY
5080 FOR AC=1 TO 15:NEXT AC
5090 REM  RESET LIGHT PEN
5100 OUT 99,0
5110 REM  TIMING LOOP TO STABILIZE DISPLAY
5120 FOR AC=1 TO 5:NEXT AC
5130 REM  STILL BEING TRIGGERED - CAN'T BE OVER THIS TARGET
          AS IT IS NOW OFF
5140 IF INP(99)>127 THEN RETURN
5150 REM  TURN TARGET ON
5160 PRINT LN;
5170 REM  STABILIZE DISPLAY/PEN
5180 FOR AC=1 TO 10:NEXT AC
5190 REM  PEN NOT TRIGGERED - CANNOT BE ON TARGET - RETURN
5200 IF INP(99)<=127 THEN RETURN
5210 REM  REPEAT TO MAKE SURE
5220 NEXT AB
5230 REM  OVER TARGET - SET STATUS VALUE
5240 P1=1
5250 REM  RESET LIGHT PEN
5260 OUT 99,0
5270 REM  RETURN TO MAIN PROGRAM
5280 RETURN
8000 REM  DATA FOR QUESTIONS - CONSISTING OF DATE FIRST,'
      FACT
          SECOND - YOU MAY OF COURSE CHANGE ANY OF THESE'
          YOU DESIRE AS LONG AS YOU HAVE FIFTY IN ALL.
8010 DATA "SEPTEMBER 6,1901","U.S. PRESIDENT WILLIAM MCKINLEY
     ASSASINATED"
8020 DATA "NOVEMBER 16,1907","STATE OF OKLAHOMA MADE THE
     46TH STATE OF THE UNION"
8030 DATA "AUGUST 15, 1912","PANAMA CANAL OPENS"
8040 DATA "JANUARY 6,1912","NEW MEXICO 46TH STATE TO ENTER
     UNION"
8050 DATA "FEBRUARY 12,1912","ARIZONA 48TH STATE TO ENTER
     UNION"
8060 DATA "MARCH 18, 1912","KING GEORGE OF GREECE ASSASINATED"
8070 DATA "JULY 28,1914","WORLD WAR I"
8080 DATA "APRIL 6, 1917","U.S.A. DECLARES WAR ON GERMANY"
8090 DATA "NOVEMBER 11, 1918","ARMISTICE DECLARED ON THE
     WESTERN FRONT"
8100 DATA "DECEMBER 30,1918","U.S.S.R. ESTABLISHED"
8110 DATA "OCTOBER 29, 1929","NEW YORK STOCK MARKET CRASH"
8120 DATA "JANUARY 30, 1933","ADOLPH HITLER BECOMES CHANCELLOR
     OF GERMANY"
8130 DATA "MARCH 27,1933","JAPAN RESIGNS FROM THE LEAGUE
     OF NATIONS"
8140 DATA "OCTOBER 14,1933","GERMANY RESIGNS FROM THE
     LEAGUE OF NATIONS"
8150 DATA "DECEMBER 11,1936","ABDICATION OF KING EDWARD
     VIII"
8160 DATA "MARCH 28, 1939","SPANISH CIVIL WAR ENDS"
8170 DATA "SEPTEMBER 3, 1939","ENGLAND & FRANCE DECLARE '
     WAR ON GERMANY"
8180 DATA "DECEMBER 8, 1941","U.S.A. DECLARES WAR ON JAPAN"
8190 DATA "JULY 16, 1945","FIRST ATOM BOMB TEST"
8200 DATA "AUGUST 6, 1945","ATOM BOMB DROPPED ON HIROSHIMA"
8210 DATA "AUGUST 15, 1945","JAPAN SURRENDERS WORLD WAR '
     II"
8220 DATA "MAY 7, 1945","GERMANY SURRENDERS - WORLD WAR '
     II"
8230 DATA "JULY 9, 1951","BRITAIN & FRANCE FORMALLY END '
     WORLD WAR II"
8240 DATA "JUNE 2, 1953","QUEEN ELIZABETH II CORONATED"
8250 DATA "SEPTEMBER 15, 1953","KHRUSHCHEV ELECTED FIRST
     SECRETARY OF U.S.S.R."
8260 DATA "NOVEMBER 3, 1957","U.S.S.R. LAUNCHES FIRST
     SATELLITE WITH LIVE DOG"
8270 DATA "JANUARY 3, 1959","ALASKA BECOMES 49TH STATE
     OF THE UNION"
8280 DATA "FEBRUARY 19, 1960","PRINCE ANDREW BORN"
8290 DATA "JANUARY 20, 1961","KENNEDY ELECTED 35TH PRESIDENT"
8300 DATA "JUNE 3, 1963","POPE JOHN XXIII DIES"
8310 DATA "NOVEMBER 22, 1963","KENNEDY ASSASINATED IN
     DALLAS"
8320 DATA "JANUARY 24, 1965","WINSTON CHURCHILL DIES"
8330 DATA "FEBRUARY 7, 1965","U.S. BEGINS BOMBING OF VIET
     NAM"
8340 DATA "SEPTEMBER 9, 1965","FRANCE WITHDRAWS FROM NATO"
8350 DATA "JULY 26, 1953","AUTHOR OF THIS ARTICLE IS BORN"
8360 DATA "MAY 15, 1957","BRITAIN ATTAINS NUCLEAR STATUS"
8370 DATA "APRIL 3, 1812","LOUISIANA BECOMES 18TH STATE '
     OF THE UNION"
8380 DATA "APRIL 11, 1814","NAPOLEON ABDICATES"
8390 DATA "JANUARY 20, 1820","GEORGE III DIES"
8400 DATA "MAY 5, 1821","NAPOLEON I DIES"
8410 DATA "SEPTEMBER 1, 1976","AUTHOR LOCKS HIMSELF OUT '
     OF HOUSE"
8420 DATA "APRIL 30, 1789","WASHINGTON ELECTED 1ST PRESIDENT"
8430 DATA "DECEMBER 2, 1859","JOHN BROWN HANGED"
8440 DATA "MARCH 4, 1861","ABRAHAM LINCOLN INAUGURATED"
8450 DATA "APRIL 14, 1865","LINCOLN ASSASSINATED"
8460 DATA "JULY 1, 1867","B.N.A. GIVEN ROYAL ASSENT"
8470 DATA "JUNE 25, 1876","CUSTER'S LAST STAND"
8480 DATA "DECEMBER 15, 1890","SITTING BULL KILLED"
8490 DATA "JULY 21, 1861","FIRST BATTLE OF BULL RUN"
8500 DATA "DECEMBER 18, 1865","U.S. ABOLISHES SLAVERY"
8510 DATA "APRIL 9, 1865","GENERAL LEE SURRENDERS - CIVIL
     WAR"
```


Reader Service—see page 226

✓Reader Service—see page 226

Learn to use the RST instruction call to greatest advantage NOW!

Get Serious

Roger L. Pape
7545 Marble Dr.
Liverpool, NY 13088

Anyone who has attempted to trace the program flow of the system software in the Radio Shack TRS-80 has probably encountered the RST (restart) instruction.

Some calls are used by the system to invoke frequently used utility routines. Details of these routines are provided so that a user might take advantage of the system software to reduce the coding required in application programs.

The RST Instruction

In the Z-80 instruction set the RST provides a compact (one byte) subroutine call. Its major advantage is a savings of two bytes over the normal CALL nnnn instruction. While the RST was intended to service interrupts, it can also be used for frequently called utility routines in the system software in order to save as much memory space as possible.

Unfortunately, its format limits RST's flexibility. A description of the RST instruction, as described in the vendor literature, is shown in Table 1.

(Note that the Z-80 assembly language convention uses the actual address for the operand, whereas 8080 code uses operands ranging from 0 to 7.)

Upon execution, a call is made to one of eight fixed memory locations (three "address" bits in instruction). These implied addresses are only eight bytes apart, allowing little room for any code of substance. Therefore, the usual practice is simply to put jump instructions, which transfer to an area where more space is available, at these locations.

The restart addresses are at page zero, the lowest memory addresses. In the TRS-80 this area is read-only memory (ROM) and cannot be modified by the user. To avoid being locked into specific addresses, the RST jumps twice, first from the page zero ROM area to an area in writeable RAM memory, which can be modified.

These standard locations in RAM (referred to as the restart vector table) are defined by the jump instructions burned into ROM and, in turn, contain jump instructions to the area where the actual service routine resides. Because of these extra jumps, it is obvious that the RST instruction has no speed advantage over the CALL instruction. As stated earlier, the RST is used in the Level II software to save memory space.

Using RST with Level II

The default locations for RST 08H through RST 38H routines are defined at system initialization (power-up), when jump instructions are loaded into RAM location 4000H to 4014H. The addresses loaded in this area can be modified after power-up to change any of the RST routines, except the power-up routine (RST 0). Note, however, that the TRS-80 Level II software makes use of six of the eight available RST calls. The user should not modify any of those six vectors, with the possible exception of the last one, RST 38H.

Table 2 shows the sequence of addresses in hexadecimal called by the various RST instructions and the actual ser-

Opcode	Operand
RST	p

1 1 ← t → 1 1 1

where t = p/8

Operation:

$(SP-1) \leftarrow PC_H$
$(SP-2) \leftarrow PC_L$
$SP \leftarrow SP-2$
$PC_H \leftarrow 0$
$PC_L \leftarrow p$

Table 1. The Format of RST

```
RST   Address sequence(hex)   Level II Routine

 0    0000:                   DI              ;disable interrupts
                              XOR A           ;clear A register
                              JP 0674H        ;to system initialization

 1    0008   4000   1C96:     LD A,(HL)       ;get current character
                              EX (SP),HL      ;get "return addr" from stack
                              CP (HL)         ;compare char to that after CALL instr
                              INC HL          ;increment pointer
                              EX (SP),HL      ;restore as adjusted return addr
                              JP Z,1D78H      ;continue as RST 10H if same
                              JP 1997H        ;else syntax error

 2    0010   4003   1D78: NXT: INC HL        ;increment string pointer
                              LD A,(HL)       ;get character
                              CP 3AH          ;compare to :
                              RET NC          ;return if equal or higher
                              CP 20H          ;check for space
                              JP A,NXT        ;skip over spaces
                              CP 08H          ;also skip
                              JR NC,CHK       ;    HT
                              CP 09H          ;    and
                              JP NC,NXT       ;    LF
                         CHK: CP 30H          ;compare to 0
                              CCF             ;compl carry flag (set if digit)
                              INC A           ;these 2 instr clear zero flag
                              DEC A           ;in case character was 0
                              RET

 3    0018   4006   1C90:     LD A,H          ;get high byte of HL pair
                              SUB D           ;subtract off high byte of DE pair
                              RET NZ          ;if not same, flags set OK
                              LD A,L          ;else compare
                              SUB E           ;low bytes
                              RET

 4    0020   4009   25D9:     LD A,(40AFH)    ;get variable type
                              CP 08H          ;double precision?
                              JR NC,DP        ;if so, jump below
                              SUB 03H         ;sets sign flag if integer
                                              ;sets zero flag if string
                              OR A            ;clears P/V flag if single
                              SCF             ;set carry flag, not double
                              RET
                         DP:  SUB 03H         ;clears zero and carry flags
                              OR A            ;sets P/V flag
                              RET

 5    0028   400C:           RET

 6    0030   400F:           RET

 7    0038   4012:           EI              ;reenable interrupts
                              RET
```

Table 2.

vice routine that is finally executed. These are the defaults initialized by the Level II software.

The following sections discuss each of these routines in more detail. If one wishes to use them in his own programs, the inputs expected and the outputs returned are described. In several cases, the A register is used within the routine and modified. If the contents of this register prior to the RST call must be preserved, either store it away (e.g. push it on the stack) or, if possible load the value into another register that is not affected.

RST 0 (00H)

This routine is invoked when the system is powered up, generating a hardware reset and initializing the program counter to 0000. A software call of RST 00H results in the same restart.

Basically, the initialization proceeds as follows. Interrupts are disabled and the cassette output port is cleared. The default RST vectors and device control blocks are copied from ROM into the lowest RAM locations, starting at 4000H. The system then checks to see whether to continue the initialization from ROM or from disk.

If a disk controller is present in the system and the break key is not depressed, it tries to copy a one-page bootstrap routine from disk. Note that because the disk controller is in the expansion interface, the break key must be depressed, to prevent the system from dropping into the disk initialization when the interface, but no disk, is present. Otherwise, the initialization con-

MODEL II

26-4002
64K 1 Drive
$3499.00

MODEL III

26-1061 4K I. $630.00
26-1062 16K III. 900.00
26-1063 32K III
2-Drives, RS232. . . . . 2246.00

AUTHORIZED TRS-80® DEALER A301

COMPUTER SPECIALISTS

26-1056 16K Level II System with Keypad. $670.00
26-1145 RS-232 Board 84.00
26-1140 "O" K Interface 249.00
26-1141 "16" K Interface 365.00
26-1142 "32" K Interface 476.00
26-1160 Mini Disk - Drive O 424.00
26-1161 Mini Disk - Additional 424.00
26-1154 Lineprinter II 720.00
26-1156 Lineprinter III 1799.00
26-1180 Voice Synthesiser 339.00
26-1181 VOXBOX 145.00
26-1104 Factory Upper/Lower
Case Modifcation Installed. 70.00
26-1506 Scripsit - Tape 60.00
26-1563 Scripsit - Disk 85.00

NOTE: Call for availability of VIDEO TEX, Model III, Color, and other new products.

MODEL I

26-1054
4K Level II
$552.00

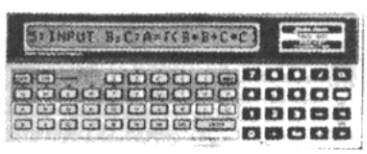
COLOR

26-3001 4K. $360.00
26-3002 16K. 540.00
26-3010 Color Video. 360.00
26-1206 Recorder. 54.00
26-3008 Joysticks. 22.50

ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES DEDUCT 10% FROM CATALOG PRICE

Novation Cat Modem . . $149.00
CCA Data Management System. 72.00
Adventure Games Games 1-9 each. 14.00

Pocket Computer



26-3501 1.9K P.C. $225.00
26-3503 Cassette I/F. 45.00
14-812 Recorder. 72.00

CENTRONICS

Fast 100 CPS Centronics
730 Printer. $675.00
Text Quality Centronics
737 Printer. $850.00

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

Acorn Software Products, Inc.

GAMES:
Alien Invasion $9.00
Stock Market 9.00
Star Trek 9.00
Block 'Em 9.00
Ting-Tong 9.00
UTILITIES:
System Savers 14.00
EDUCATION:
Language Teacher 18.00

FREE: PRICE LIST UPON REQUEST

1-800-841-0860 Toll Free Order Entry

✓72

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

Full Factory Warranty on All Items Sold.

Largest Inventory In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

tinues from the Level II ROM software, copying additional default values from ROM to RAM, setting various memory pointers and beginning the MEMORY SIZE? dialog.

RST 1 (08H)

The RST 08H call checks whether the last character read from an input string is a specific character required by the syntax of the statement being processed. If the proper character is found, the routine continues to the next character in the same manner as the RST 10H call (see below); otherwise a syntax error return is used. The character to be checked must be stored in the next byte after this instruction.

Inputs:	HL register pair points to character to be checked (requires character immediately after instruction).
Outputs:	a) If proper character, A contains next character not a space, HT, or LF. HL points to location of character returned.
Flags:	Carry flag set if digit (0 to 9), else cleared; zero flag set if colon (:), else cleared.
	b) If wrong character, jumps to SN error return.
Other registers modified:	If error return, A,B,C,E.

Table 3.

Inputs:	HL points one location *prior* to next character to check.
Outputs:	A contains next character, not a space, HT, or LF. HL points to location of character returned. Flags: Carry flag set if digit (0 to 9), else cleared; zero flag set if colon (:), else cleared. Other registers modified: None.

Table 4.

✓Reader Service—see page 226

Inputs:	HL and DE contain 16 bit values to be compared.	
Outputs:	Condition	Appropriate flags to check
	Address in HL>DE	Carry flag cleared, zero flag cleared
	Address in HL<DE	Carry flag set
	Signed val in HL>DE	[Sign flag ⊕ P/V flag] = 0, Zero flag clear
	Signed val in HL<DE	[Sign flag ⊕ P/V flag] = 1
	HL = DE	Zero flag set, carry and sign flags clear
	Other registers modified: A (Note: HL and DE not modified.)	

Table 5.

As an example of the use of this routine, suppose the text just processed *must* be followed by a comma and then some additional data. The code would be:

```
RST 10H
DEFB','
```

Note that the RST routine automatically increments the return address past the defined character.

RST 2 (10H)

This service routine is used primarily for scanning input character strings. It retrieves the next non-blank character after the current HL pointer location, also skipping over any horizontal tab (HT) and line feed (LF) characters that may be in the string.

In addition to returning the character and its memory location, it also sets or clears certain condition flags to indicate the type of character retrieved. The RST 10H call is usually followed by a conditional jump, depending on the type of character expected next. (Table 4).

Be sure to note that the HL register pair is incremented *before* the character is read. As an example of its use, a routine may be converting digits in the input string to an integer value. The process would be terminated if a non-digit character is encountered. Then the coding would include:

```
RST 10H
RET NC
:
(process character in A)
:
```

RST 3 (18H)

The function of this routine compares the contents of the HL register pair to the contents of the DE register pair. It performs an HL - DE compare and sets the appropriate condition flags.

The routine is particularly useful when comparing a memory pointer to some memory limit or other stored pointer. When comparing memory addresses, the carry flag should be tested rather than the sign flag.

An address greater than 32K is considered higher than one below it, but the sign bit is set (negative) for these upper addresses. Comparisons across the 32K boundary could cause erroneous results if the sign flag is used.

On the other hand, if the RST call is used for a true arithmetic comparison, use the sign bit instead. In the latter case, the sign bit must be exclusive ORed with the overflow flag in case the difference exceeds the 16 bit number range. (Table 5).

For example, to generate a jump if the HL pointer is higher than or the same as the DE pointer, the code would be:

```
RST 18H
JP NC,nnnn
```

RST 4 (20H)

This service routine is used in the Level II software to check the type of the variable that is currently being processed. The variable type indicator (corresponding to byte length) is stored in location 40AFH of the reserved area in RAM. (Table 6).

The routine tests the type indicator in such a way that one of the four condition flags used has a unique setting for each of the types. (Table 7).

A RST 20H call is followed by a conditional jump instruction. For example, to branch if the variable is single precision, the code would be:

```
RST 20H
JP PO,nnnn
```

RST 5 (28H)

Not used by Level II BASIC; therefore, the default is an immediate return. This RST is executed in the Level II keyboard driver when the break key is depressed. Modifying the restart vector (400C-E) allows one to intercept execution when the break key is pressed.

RST 6 (30H)

Not used by Level II BASIC: therefore, the default is an immediate return.

RST 7 (38H)

The default coding for this RST simply reenables interrupts and returns. One should be aware that this RST is somewhat special.

First of all, the machine code for this instruction is FFH, a value to which much of memory initializes. An inadvertent jump to one of those locations will invoke the restart.

Secondly, if the Z-80 processor is set in interrupt Mode 1 by the IM1 instruction, a nonmaskable interrupt (the kind generated by pushing the reset button behind the keyboard) executes a restart to location 38H instead of the normal NMI restart to location 66H when in Mode 0 (the power-up default). Setting interrupt Mode 1 and loading a jump instruction at locations 4012-4H allows one to intercept execution after the reset button is pushed, rather than enter the reinitialization routine in ROM.

Conclusion

The service routines called by the RST instruction in the TRS-80 Level II software were chosen because they are frequently used functions. It is to the advantage of any serious assembly language programmer to understand these routines and try to use them.

For example, a program which reads in and scans lines of text might be written to utilize the RST 10H call effectively. The programmer who is interested in the ultimate compactness might also consider using those RST's not claimed by the system software.

For DOS users, be sure to first check how your operating system uses them. ■

Variable type	(40AFH)
Integer	2
String	3
Single precision	4
Double precision	8

Table 6.

Inputs:	(40AFH) = variable type
Outputs:	Flags—Sign flag set only if integer, else cleared;
	zero flag set only if string, else cleared;
	P/V flag cleared (P0) only if single prec, else set (PE)
	Carry flag cleared only if double precision, else set
	Other registers modified: A (contains type indicator – 3.)

Table 7.

PMC-80

Level II 16K at $645

SOFTWARE COMPATIBLE

- Reads all Level II BASIC tapes
- Reads all SYSTEM tapes
- Full range of peripherals
- Video output for monitor and TV
- Optional FASTLOAD at 8000 band
- Optional Upper/Lower case

The PMC-80 is a "work-alike" computer to the popular TRS-80* Model I, Level II by Tandy, Radio Shack. The PMC-80 has 16K bytes of RAM and the complete Level II 12K BASIC ROM by Microsoft that makes it 100% software compatible with programs from Radio Shack and from the hundreds of other independent suppliers. The built-in cassette player reads standard Radio Shack programs for the TRS-80.*

The PMC-80 will operate with any of the many peripherals Radio Shack and other independent vendors have invented to plug into the TRS-80.* Most importantly, the Interface Adapter permits Expansion Interfaces with memory expansion to 48K to be added. An Expansion Interface will also permit the addition of Radio Shack compatible 5¼" disks and disk operating systems, RS 232, printers, etc.

Sold through computer stores.

*TRS-80 is a registered trademark of Tandy, Radio Shack.

✓422

Personal Micro Computers, Inc.

475 Ellis Street, Mountain View, CA 94043 (415) 962-0220

More on this handy speed-up device for the Shack's Level II BASIC.

The Useful USR(0) Function

Terry Kepner
PO Box 481
Peterborough, NH 03458

As I slowly worked my way into assembly language programming, I discovered the usefulness of the Radio Shack USR(0) function. To those of you who have not yet tripped across this utility, USR(0) allows you to call assembly language routines from BASIC programs. This is handy when you want to speed up the normally tedious graphic functions and string manipulations of Level II BASIC.

There are two primary methods used to imbed these routines into your program. The first method is the easiest for a programmer to use, but can lead to problems for the end user. It involves loading the assembly language routine into memory via the System command, and then using CLOAD for the BASIC part of the program.

Second Method

The second method takes advantage of POKE and lets you load the assembly language routine as a part of the BASIC program itself. To do this you place the assembly language instructions in data statements, then POKE these statements into active memory.

Since the first method requires loading two different programs, user errors may creep into the programs. The second method eliminates this problem,

Program Listing

```
4A00            00010          ORG     4A00H
                00020 ;*************************
                00030 ;** TBUG+  JAN. 01,1980 **
                00040 ;*************************
43DD            00050 RETURN   EQU     43DDH
3C00            00060 VIDEO    EQU     3C00H
0001            00070 COUNT    DEFS    1
0001            00080 BLOCK    DEFS    1
0002            00090 SVMEMA   DEFS    2
0002            00100 SVSCRA   DEFS    2
                00110
4A06 FE43       00120 START    CP      'C'
4A08 2811       00130          JR      Z,CLS
                00140
4A0A FE44       00150          CP      'D'
4A0C 2827       00160          JR      Z,DUMP
                00170
4A0E FE46       00180          CP      'F'
4A10 CA0D48     00190          JP      Z,480DH ;REP FIX RTN CHECK
                00200
4A13 FE53       00210          CP      'S'
4A15 CA014B     00220          JP      Z,SHIFT
                00230
                00240 ;INSERT ADDIT'L COMMANDS HERE.......
                00250
4A18 C3EA43     00260          JP      43EAH ;RET-NO HIT
                00270
                00280 ;*************************
                00290 ;** CLEAR SCREEN ROUTINE **
                00300 ;*************************
4A1B CD274A     00310 CLS      CALL    CLRSCR
                00320
4A1E 21003C     00330          LD      HL,VIDEO
4A21 223D48     00340          LD      (483DH),HL
                00350
4A24 C3DD43     00360          JP      RETURN
                00370
4A27 21003C     00380 CLRSCR   LD      HL,VIDEO
4A2A 11013C     00390          LD      DE,VIDEO+1
4A2D 010004     00400          LD      BC,400H
4A30 3620       00410          LD      (HL),20H
4A32 EDB0       00420          LDIR
4A34 C9         00430          RET
                00440
                00450 ;*************************
                00460 ;** DUMP MEMORY ROUTINE  **
                00470 ;*************************
                00480
4A35 CD3245     00490 DUMP     CALL    4532H
4A38 CD8945     00500          CALL    4589H
4A3B 324848     00510          LD      (4848H),A
                00520
4A3E CD8945     00530          CALL    4589H
4A41 323F48     00540          LD      (483FH),A
                00550
4A44 CD274A     00560          CALL    CLRSCR
                00570
4A47 DD2A3F48   00580          LD      IX,(483FH) ;MEM LOCN
4A4B FD21003C   00590          LD      IY,VIDEO ;SCR LOCN
4A4F 01D000     00600          LD      BC,208
                00610
4A52 AF         00620          XOR     A
4A53 32004A     00630          LD      (COUNT),A
4A56 32014A     00640          LD      (BLOCK),A
4A59 DD22024A   00650          LD      (SVMEMA),IX
4A5D FD22044A   00660          LD      (SVSCRA),IY
4A61 CD774B     00670          CALL    ADDRSS
                00680
4A64 FD23       00690          INC     IY
                00700
4A66 DD7E00     00710 DSPDMP   LD      A,(IX)
                00720
4A69 CD674B     00730          CALL    DSPRTN
                00740
                00750 ;ASCII CODE RTN
                00760
4A6C FDE5       00770          PUSH    IY
                00780
4A6E DDE5       00790          PUSH    IX      ;CURR MEM PTR
4A70 E1         00800          POP     HL
                00810
4A71 ED5B024A   00820          LD      DE,(SVMEMA)
4A75 B7         00830          OR      A
4A76 ED52       00840          SBC     HL,DE
                00850
4A78 ED5B044A   00860          LD      DE,(SVSCRA)
4A7C 19         00870          ADD     HL,DE
                00880
4A7D 112E00     00890          LD      DE,2EH
4A80 19         00900          ADD     HL,DE
                00910
4A81 E5         00920          PUSH    HL
4A82 FDE1       00930          POP     IY
                00940
4A84 3E2E       00950          LD      A,'.'
4A86 FD7700     00960          LD      (IY),A
                00970
4A89 DD7E00     00980          LD      A,(IX)
                00990
4A8C FE30       01000          CP      30H
4A8E FA994A     01010          JP      M,ASCEND
                01020
4A91 FE5B       01030          CP      5BH
4A93 F2994A     01040          JP      P,ASCEND
                01050
4A96 FD7700     01060          LD      (IY),A
                01070
4A99 FDE1       01080 ASCEND   POP     IY
                01090
4A9B DD23       01100          INC     IX
4A9D 0B         01110          DEC     BC
                01120
4A9E 3A004A     01130          LD      A,(COUNT)
4AA1 3C         01140          INC     A
4AA2 32004A     01150          LD      (COUNT),A
4AA5 FE04       01160          CP      4
4AA7 2803       01170          JR      Z,SPACE
4AA9 C3664A     01180          JP      DSPDMP
                01190
4AAC FD23       01200 SPACE    INC     IY
4AAE AF         01210          XOR     A
4AAF 32004A     01220          LD      (COUNT),A  ;CLEAR  COUNT
                01230
4AB2 3A014A     01240          LD      A,(BLOCK)
4AB5 3C         01250          INC     A
4AB6 32014A     01260          LD      (BLOCK),A
                01270
4AB9 FE04       01280          CP      4
```

Program continues

```
4ABB 28A9      01290          JR      NZ,DSPDMP
               01300
4ABD AF        01310 NEWLIN   XOR     A
4ABE 32014A    01320          LD      (BLOCK),A  ;CLEAR BLOCK
               01330
4AC1 111700    01340          LD      DE,0017H
4AC4 FD19      01350          ADD     IY,DE
4AC6 FD22044A  01360          LD      (SVSCRA),IY
4ACA DD22024A  01370          LD      (SVMEMA),IX
4ACE 78        01380          LD      A,B
4ACF B1        01390          OR      C
4AD0 2807      01400          JR      Z,DSPRET
               01410
4AD2 CD774B    01420          CALL    ADDRSS
               01430
4AD5 FD23      01440          INC     IY
4AD7 188D      01450          JR      DSPDMP
               01460
4AD9 21803F    01470 DSPRET   LD      HL,3F80H
4ADC 223D48    01480          LD      (483DH),HL
4ADF C3DD43    01490          JP      RETURN
               01500
4AE2 4F        01510 DSPCHR   LD      C,A
4AE3 CB3F      01520          SRL     A
4AE5 CB3F      01530          SRL     A
4AE7 CB3F      01540          SRL     A
4AE9 CB3F      01550          SRL     A
4AEB CDF74A    01560          CALL    CHECK
4AEE 67        01570          LD      H,A
4AEF 79        01580          LD      A,C
4AF0 E60F      01590          AND     0FH
4AF2 CDF74A    01600          CALL    CHECK
4AF5 6F        01610          LD      L,A
4AF6 C9        01620          RET
4AF7 C630      01630 CHECK    ADD     A,30H
4AF9 FE3A      01640          CP      3AH
4AFB FA004B    01650          JP      M,CHECK1
4AFE C607      01660          ADD     A,7
4B00 C9        01670 CHECK1   RET
               01680
               01690 ;*************************
               01700 ;** MOVE UP/DOWN  ROUTNE **
               01710 ;*************************
               01720
4B01 CD3245    01730 SHIFT    CALL    4532H
               01740
4B04 CD8945    01750          CALL    4589H
4B07 324248    01760          LD      (4842H),A       ;SOURCE-MSB
               01770
4B0A CD8945    01780          CALL    4589H
4B0D 324148    01790          LD      (4841H),A       ;SOURCE-LSB
               01800
4B10 CD7B45    01810          CALL    457BH
               01820
4B13 CD8945    01830          CALL    4589H           ;DEST-MSB
4B16 324448    01840          LD      (4844H),A
               01850
4B19 CD8945    01860          CALL    4589H
4B1C 324348    01870          LD      (4843H),A       ;DEST-LSB
               01880
4B1F CD7B45    01890          CALL    457BH
               01900
4B22 CD8945    01910          CALL    4589H
4B25 324648    01920          LD      (4846H),A       ;#BYTES - MSB
               01930
4B28 CD8945    01940          CALL    4589H
4B2B 324548    01950          LD      (4845H),A       ;#BYTES - LSB
               01960
4B2E 2A4148    01970          LD      HL,(4841H)
4B31 ED5B4348  01980          LD      DE,(4843H)
4B35 B7        01990          OR      A       ;RESET CARRY
4B36 ED52      02000          SBC     HL,DE
4B38 F24D4B    02010          JP      P,MOVDWN
               02020
4B3B CD554B    02030 MOVUP    CALL    MSETUP
4B3E E5        02040          PUSH    HL
4B3F D5        02050          PUSH    DE
4B40 E1        02060          POP     HL
4B41 09        02070          ADD     HL,BC
4B42 E5        02080          PUSH    HL
4B43 D1        02090          POP     DE
4B44 E1        02100          POP     HL
4B45 09        02110          ADD     HL,BC
4B46 2B        02120          DEC     HL
4B47 1B        02130          DEC     DE
4B48 EDB8      02140          LDDR
4B4A C3DD43    02150          JP      RETURN
4B4D CD554B    02170 MOVDWN   CALL    MSETUP
               02180
4B50 EDB0      02190          LDIR
               02200
4B52 C3DD43    02210          JP      RETURN
4B55 21C03F    02220 MSETUP   LD      HL,3FC0H
4B58 223D48    02230          LD      (483DH),HL
               02240
4B5B 2A4148    02250          LD      HL,(4841H)
4B5E ED5B4348  02260          LD      DE,(4843H)
4B62 ED4B4548  02270          LD      BC,(4845H)
               02280
4B66 C9        02290          RET
               02300
4B67 C5        02310 DSPRTN   PUSH    BC
4B68 CDE24A    02320          CALL    DSPCHR
4B6B C1        02330          POP     BC
               02340
4B6C FD7400    02350          LD      (IY),H
4B6F FD23      02360          INC     IY
               02370
4B71 FD7500    02380          LD      (IY),L
4B74 FD23      02390          INC     IY
               02400
4B76 C9        02410          RET
               02420
4B77 DDE5      02430 ADDRSS   PUSH    IX
4B79 D1        02440          POP     DE
               02450
4B7A 7A        02460          LD      A,D
4B7B CD674B    02470          CALL    DSPRTN
               02480
4B7E 7B        02490          LD      A,E
4B7F CD674B    02500          CALL    DSPRTN
```


↙Reader Service—see page 226

DEC.	HEX	ASCII	DEC.	HEX	ASCII
32	20	SPACE	80	50	P
33	21	!	81	51	Q
34	22	"	82	52	R
35	23	#	83	53	S
36	24	$	84	54	T
37	25	%	85	55	U
38	26	&	86	56	V
39	27	'	87	57	W
40	28	(	88	58	X
41	29	)	89	59	Y
42	2A	*	90	5A	Z
43	2B	+	91	5B	↑
44	2C	,	92	5C	↓
45	2D	-	93	5D	←
46	2E	.	94	5E	→
47	2F	/	95	5F	_
48	30	0	96	60	@
49	31	1	97	61	a
50	32	2	98	62	b
51	33	3	99	63	c
52	34	4	100	64	d
53	35	5	101	65	e
54	36	6	102	66	f
55	37	7	103	67	g
56	38	8	104	68	h
57	39	9	105	69	i
58	3A	:	106	6A	j
59	3B	;	107	6B	k
60	3C	<	108	6C	l
61	3D	=	109	6D	m
62	3E	>	110	6E	n
63	3F	?	111	6F	o
64	40	@	112	70	p
65	41	A	113	71	q
66	42	B	114	72	r
67	43	C	115	73	s
68	44	D	116	74	t
69	45	E	117	75	u
70	46	F	118	76	v
71	47	G	119	77	w
72	48	H	120	78	x
73	49	I	121	79	y
74	4A	J	122	7A	z
75	4B	K	123	7B	↑
76	4C	L	124	7C	↓
77	4D	M	125	7D	←
78	4E	N	126	7E	→
79	4F	O	127	7F	—

Character Codes Decimal & Hexadecimal

but introduces its own: conversion.

BASIC uses decimal numbers when POKEing and assembly language uses hexadecimal numbers for its instructions.

This means that the programmer must convert the hexadecimal numbers into decimal in order to use the POKE process. This is a painful and boring procedure. There is also a good possibility that the programmer will make an error.

It didn't take me long—three Opcodes—to realize that doing it that way was for the birds.

So, I wrote a short program that did this chore for me and also produced a hard copy for future use.

Since I have found this hard copy indispensable, I've decided to share it with the rest of computerdom. The rest of this article is the Z-80 Opcode Hexadecimal-to-Decimal Conversion Chart. Also included is a hex-decimal listing of the ASCII codes, including both upper and lowercase characters. (The new Radio Shack manual leaves out the lowercase letters.)

I hope you find this chart to be as useful as I find it. ■

Hexadecimal to Decimal Conversion Chart

```
00          0
018405      1, 132, 5
02          2
03          3
04          4
05          5
0620        6, 32
```

Hexadecimal	Decimal
07	7
08	8
09	9
0A	10
0B	11
0C	12
0D	13
0E20	14, 32
0F	15
102E	16, 46
118405	17, 132, 5
12	18
13	19
14	20
15	21
1620	22, 32
17	23
182E	24, 46
19	25
1A	26
1B	27
1C	28
1D	29
1E20	30, 32
1F	31
202E	32, 46
218405	33, 132, 5
228405	34, 132, 5
23	35
24	36
25	37
2620	38, 32
27	39
282E	40, 46
29	41
2A8405	42, 132, 5
2B	43
2C	44
2D	45
2E20	46, 32
2F	47
302E	48, 46
318405	49, 132, 5
328405	50, 132, 5
33	51
34	52
35	53
3620	54, 32
37	55
382E	56, 46
39	57
3A8405	58, 132, 5
3B	59
3C	60
3D	61
3E20	62, 32
3F	63
40	64
41	65
42	66
43	67
44	68
45	69
46	70
47	71
48	72
49	73
4A	74
4B	75
4C	76
4D	77
4E	78
4F	79
50	80
51	81
52	82
53	83
54	84
55	85
56	86
57	87
58	88
59	89
5A	90
5B	91
5C	92
5D	93
5E	94
5F	95
60	96
61	97
62	98
63	99
64	100
65	101
66	102
67	103
68	104
69	105
6A	106
6B	107
6C	108
6D	109
6E	110
6F	111
70	112
71	113
72	114
73	115
74	116
75	117
76	118
77	119
78	120
79	121
7A	122

Hexadecimal	Decimal
7B	123
7C	124
7D	125
7E	126
7F	127
80	128
81	129
82	130
83	131
84	132
85	133
86	134
87	135
88	136
89	137
8A	138
8B	139
8C	140
8D	141
8E	142
8F	143
90	144
91	145
92	146
93	147
94	148
95	149
96	150
97	151
98	152
99	153
9A	154
9B	155
9C	156
9D	157
9E	158
9F	159
A0	160
A1	161
A2	162
A3	163
A4	164
A5	165
A6	166
A7	167
A8	168
A9	169
AA	170
AB	171
AC	172
AD	173
AE	174
AF	175
B0	176
B1	177
B2	178
B3	179
B4	180
B5	181
B6	182
B7	183
B8	184
B9	185
BA	186
BB	187
BC	188
BD	189
BE	190
BF	191
C0	192
C1	193
C28405	194, 132, 5
C38405	195, 132, 5
C48405	196, 132, 5
C5	197
C620	198, 32
C7	199
C8	200
C9	201
CA8405	202, 132, 5
CB00	203, 0
CB01	203, 1
CB02	203, 2
CB03	203, 3
CB04	203, 4
CB05	203, 5
CB06	203, 6
CB07	203, 7
CB08	203, 8
CB09	203, 9
CB0A	203, 10
CB0B	203, 11
CB0C	203, 12
CB0D	203, 13
CB0E	203, 14
CB0F	203, 15
CB10	203, 16
CB11	203, 17
CB12	203, 18
CB13	203, 19
CB14	203, 20
CB15	203, 21
CB16	203, 22
CB17	203, 23
CB18	203, 24
CB19	203, 25
CB1A	203, 26
CB1B	203, 27
CB1C	203, 28
CB1D	203, 29
CB1E	203, 30
CB1F	203, 31
CB20	203, 32

Program continues

Hexadecimal	Decimal
CB21	203, 33
CB22	203, 34
CB23	203, 35
CB24	203, 36
CB25	203, 37
CB26	203, 38
CB27	203, 39
CB28	203, 40
CB29	203, 41
CB2A	203, 42
CB2B	203, 43
CB2C	203, 44
CB2D	203, 45
CB2E	203, 46
CB2F	203, 47
CB38	203, 56
CB39	203, 57
CB3A	203, 58
CB3B	203, 59
CB3C	203, 60
CB3D	203, 61
CB3E	203, 62
CB3F	203, 63
CB40	203, 64
CB41	203, 65
CB42	203, 66
CB43	203, 67
CB44	203, 68
CB45	203, 69
CB46	203, 70
CB47	203, 71
CB48	203, 72
CB49	203, 73
CB4A	203, 74
CB4B	203, 75
CB4C	203, 76
CB4D	203, 77
CB4E	203, 78
CB4F	203, 79
CB50	203, 80
CB51	203, 81
CB52	203, 82
CB53	203, 83
CB54	203, 84
CB55	203, 85
CB56	203, 86
CB57	203, 87
CB58	203, 88
CB59	203, 89
CB5A	203, 90
CB5B	203, 91
CB5C	203, 92
CB5D	203, 93
CB5E	203, 94
CB5F	203, 95
CB60	203, 96
CB61	203, 97
CB62	203, 98
CB63	203, 99
CB64	203, 100
CB65	203, 101
CB66	203, 102
CB67	203, 103
CB68	203, 104
CB69	203, 105
CB6A	203, 106
CB6B	203, 107
CB6C	203, 108
CB6D	203, 109
CB6E	203, 110
CB6F	203, 111
CB70	203, 112
CB71	203, 113
CB72	203, 114
CB73	203, 115
CB74	203, 116
CB75	203, 117
CB76	203, 118
CB77	203, 119
CB78	203, 120
CB79	203, 121
CB7A	203, 122
CB7B	203, 123
CB7C	203, 124
CB7D	203, 125
CB7E	203, 126
CB7F	203, 127
CB80	203, 128
CB81	203, 129
CB82	203, 130
CB83	203, 131
CB84	203, 132
CB85	203, 133
CB86	203, 134
CB87	203, 135
CB88	203, 136
CB89	203, 137
CB8A	203, 138
CB8B	203, 139
CB8C	203, 140
CB8D	203, 141
CB8E	203, 142
CB8F	203, 143
CB90	203, 144
CB91	203, 145
CB92	203, 146
CB93	203, 147
CB94	203, 148
CB95	203, 149
CB96	203, 150
CB97	203, 151
CB98	203, 152
CB99	203, 153
CB9A	203, 154
CB9B	203, 155
CB9C	203, 156
CB9D	203, 157
CB9E	203, 158
CB9F	203, 159
CBA0	203, 160
CBA1	203, 161
CBA2	203, 162
CBA3	203, 163
CBA4	203, 164
CBA5	203, 165
CBA6	203, 166
CBA7	203, 167
CBA8	203, 168
CBA9	203, 169
CBAA	203, 170
CBAB	203, 171
CBAC	203, 172
CBAD	203, 173
CBAE	203, 174
CBAF	203, 175
CBB0	203, 176
CBB1	203, 177
CBB2	203, 178
CBB3	203, 179
CBB4	203, 180
CBB5	203, 181
CBB6	203, 182
CBB7	203, 183
CBB8	203, 184
CBB9	203, 185
CBBA	203, 186
CBBB	203, 187
CBBC	203, 188
CBBD	203, 189
CBBE	203, 190
CBBF	203, 191
CBC0	203, 192
CBC1	203, 193
CBC2	203, 194
CBC3	203, 195
CBC4	203, 196
CBC5	203, 197
CBC6	203, 198
CBC7	203, 199
CBC8	203, 200
CBC9	203, 201
CBCA	203, 202
CBCB	203, 203
CBCC	203, 204
CBCD	203, 205
CBCE	203, 206
CBCF	203, 207
CBD0	203, 208
CBD1	203, 209
CBD2	203, 210
CBD3	203, 211
CBD4	203, 212
CBD5	203, 213
CBD6	203, 214
CBD7	203, 215
CBD8	203, 216
CBD9	203, 217
CBDA	203, 218
CBDB	203, 219
CBDC	203, 220
CBDD	203, 221
CBDE	203, 222
CBDF	203, 223
CBE0	203, 224
CBE1	203, 225
CBE2	203, 226
CBE3	203, 227
CBE4	203, 228
CBE5	203, 229
CBE6	203, 230
CBE7	203, 231
CBE8	203, 232
CBE9	203, 233
CBEA	203, 234
CBEB	203, 235
CBEC	203, 236
CBED	203, 237
CBEE	203, 238
CBEF	203, 239
CBF0	203, 240
CBF1	203, 241
CBF2	203, 242
CBF3	203, 243
CBF4	203, 244
CBF5	203, 245
CBF6	203, 246
CBF7	203, 247
CBF8	203, 248
CBF9	203, 249
CBFA	203, 250
CBFB	203, 251
CBFC	203, 252
CBFD	203, 253
CBFE	203, 254
CBFF	203, 255
CC8405	204, 132, 5
CD8405	205, 132, 5
CE20	206, 32
CF	207
D0	208
D1	209
D28405	210, 132, 5
D320	211, 32
D48405	212, 132, 5
D5	213
D620	214, 32
D7	215
D8	216
D9	217

✓Reader Service—see page 226

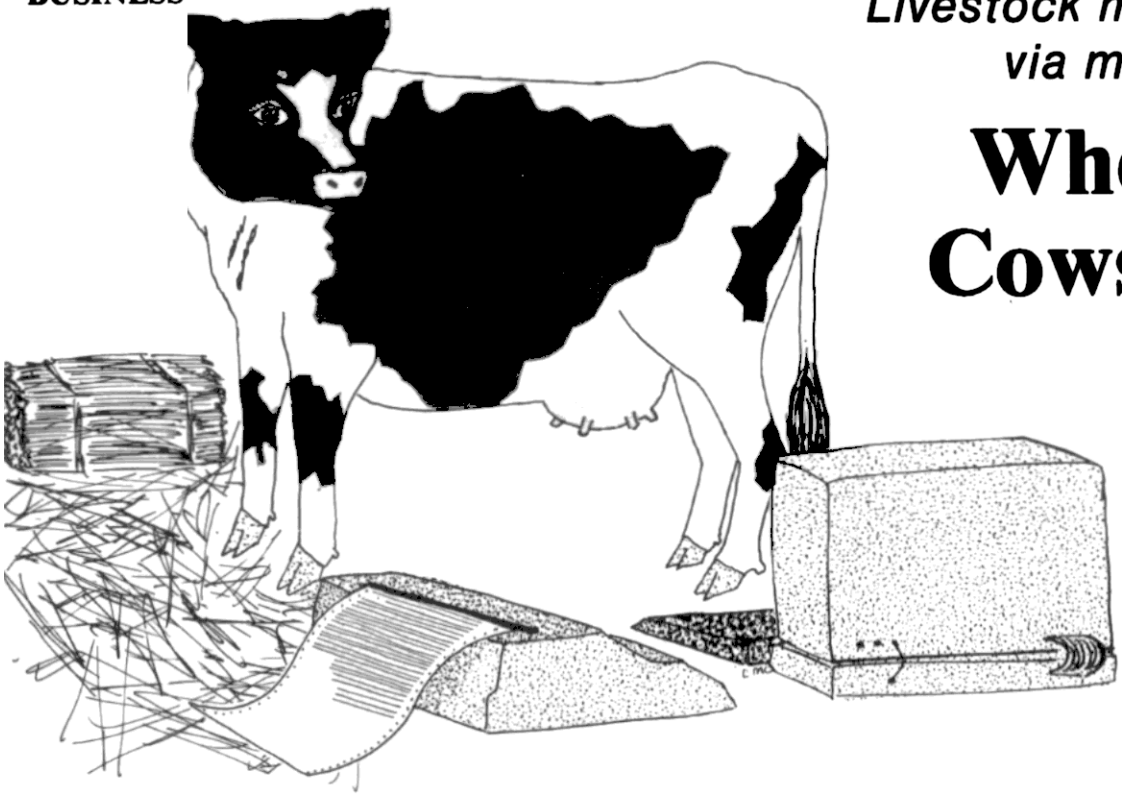
Hexadecimal	Decimal
DA8405	218, 132, 5
DB20	219, 32
DC8405	220, 132, 5
DD09	221, 9
DD19	221, 25
DD218405	221, 33, 132, 5
DD228405	221, 34, 132, 5
DD23	221, 35
DD29	221, 41
DD2A8405	221, 42, 132, 5
DD2B	221, 43
DD3405	221, 52, 5
DD3505	221, 53, 5
DD360520	221, 54, 5, 32
DD39	221, 57
DD4605	221, 70, 5
DD4E05	221, 78, 5
DD5605	221, 86, 5
DD5E05	221, 94, 5
DD6605	221, 102, 5
DD6E05	221, 110, 5
DD7005	221, 112, 5
DD7105	221, 113, 5
DD7205	221, 114, 5
DD7305	221, 115, 5
DD7405	221, 116, 5
DD7505	221, 117, 5
DD7705	221, 119, 5
DD7E05	221, 126, 5
DD8605	221, 134, 5
DD8E05	221, 142, 5
DD9605	221, 150, 5
DD9E05	221, 158, 5
DDA605	221, 166, 5
DDAE05	221, 174, 5
DDB605	221, 182, 5
DDBE05	221, 190, 5
DDCB0506	221, 203, 5, 6
DDCB050E	221, 203, 5, 14
DDCB0516	221, 203, 5, 22
DDCB051E	221, 203, 5, 30
DDCB0526	221, 203, 5, 38
DDCB052E	221, 203, 5, 46
DDCB053E	221, 203, 5, 62
DDCB0546	221, 203, 5, 70
DDCB054E	221, 203, 5, 78
DDCB0556	221, 203, 5, 86
DDCB055E	221, 203, 5, 94
DDCB0566	221, 203, 5, 102
DDCB056E	221, 203, 5, 110
DDCB0576	221, 203, 5, 118
DDCB057E	221, 203, 5, 126
DDCB0586	221, 203, 5, 134
DDCB058E	221, 203, 5, 142
DDCB0596	221, 203, 5, 150
DDCB059E	221, 203, 5, 158
DDCB05A6	221, 203, 5, 166
DDCB05AE	221, 203, 5, 174
DDCB05B6	221, 203, 5, 182
DDCB05BE	221, 203, 5, 190
DDCB05C6	221, 203, 5, 198
DDCB05CE	221, 203, 5, 206
DDCB05D6	221, 203, 5, 214
DDCB05DE	221, 203, 5, 222
DDCB05E6	221, 203, 5, 230
DDCB05EE	221, 203, 5, 238
DDCB05F6	221, 203, 5, 246
DDCB05FE	221, 203, 5, 254
DDE1	221, 225
DDE3	221, 227
DDE5	221, 229
DDE9	221, 233
DDF9	221, 249
DE20	222, 32
DF	223
E0	224
E1	225
E28405	226, 132, 5
E3	227
E48405	228, 132, 5
E5	229
E620	230, 32
E7	231
E8	232
E9	233
EA8405	234, 132, 5
EB	235
EC8405	236, 132, 5
ED40	237, 64
ED41	237, 65
ED42	237, 66
ED438405	237, 67, 132, 5
ED44	237, 68
ED4455	237, 68, 85
ED46	237, 70
ED47	237, 71
ED48	237, 72
ED49	237, 73
ED4A	237, 74
ED4B8405	237, 75, 132, 5
ED4D	237, 77
ED4F	237, 79
ED50	237, 80
ED51	237, 81
ED52	237, 82
ED538405	237, 83, 132, 5
ED56	237, 86
ED57	237, 87
ED58	237, 88
ED59	237, 89
ED5A	237, 90
ED5B8405	237, 91, 132, 5
ED5E	237, 94
ED5F	237, 95
ED60	237, 96
ED61	237, 97

Hexadecimal	Decimal
ED62	237, 98
ED67	237, 103
ED68	237, 104
ED69	237, 105
ED6A	237, 106
ED6F	237, 111
ED72	237, 114
ED738405	237, 115, 132, 5
ED78	237, 120
ED79	237, 121
ED7A	237, 122
ED7B8405	237, 123, 132, 5
EDA0	237, 160
EDA1	237, 161
EDA2	237, 162
EDA3	237, 163
EDA8	237, 168
EDA9	237, 169
EDAA	237, 170
EDAB	237, 171
EDB0	237, 176
EDB1	237, 177
EDB2	237, 178
EDB3	237, 179
EDB8	237, 184
EDB9	237, 185
EDBA	237, 186
EDBB	237, 187
EE20	238, 32
EF	239
F0	240
F1	241
F28405	242, 132, 5
F3	243
F48405	244, 132, 5
F5	245
F620	246, 32
F7	247
F8	248
F9	249
FA8405	250, 132, 5
FB	251
FC8405	252, 132, 5
FD09	253, 9
FD19	253, 25
FD218405	253, 33, 132, 5
FD228405	253, 34, 132, 5
FD23	253, 35
FD29	253, 41
FD2A8405	253, 42, 132, 5
FD2B	253, 43
FD3405	253, 52, 5
FD3505	253, 53, 5
FD360520	253, 54, 5, 32
FD39	253, 57
FD4605	253, 70, 5
FD4E05	253, 78, 5
FD5605	253, 86, 5
FD5E05	253, 94, 5
FD6605	253, 102, 5
FD6E05	253, 110, 5
FD7005	253, 112, 5
FD7105	253, 113, 5
FD7205	253, 114, 5
FD7305	253, 115, 5
FD7405	253, 116, 5
FD7505	253, 117, 5
FD7705	253, 119, 5
FD7E05	253, 126, 5
FD8605	253, 134, 5
FD8E05	253, 142, 5
FD9605	253, 150, 5
FD9E05	253, 158, 5
FDA605	253, 166, 5
FDAE05	253, 174, 5
FDB605	253, 182, 5
FDBE05	253, 190, 5
FDCB0506	253, 203, 5, 6
FDCB050E	253, 203, 5, 14
FDCB0516	253, 203, 5, 22
FDCB051E	253, 203, 5, 30
FDCB0526	253, 203, 5, 38
FDCB052E	253, 203, 5, 46
FDCB053E	253, 203, 5, 62
FDCBo546	253, 203, 5, 70
FDCB054E	253, 203, 5, 78
FDCB0556	253, 203, 5, 86
FDCB055E	253, 203, 5, 94
FDCB0566	253, 203, 5, 102
FDCB056E	253, 203, 5, 110
FDCB0576	253, 203, 5, 118
FDCB057E	253, 203, 5, 126
FDCB0586	253, 203, 5, 134
FDCB058E	253, 203, 5, 142
FDCB0596	253, 203, 5, 150
FDCB059E	253, 203, 5, 158
FDCB05A6	253, 203, 5, 166
FDCB05AE	253, 203, 5, 174
FDCB05B6	253, 203, 5, 182
FDCB05BE	253, 203, 5, 190
FDCB05C6	253, 203, 5, 198
FDCB05CE	253, 203, 5, 206
FDCB05D6	253, 203, 5, 214
FDCB05DE	253, 203, 5, 222
FDCB05E6	253, 203, 5, 230
FDCB05EE	253, 203, 5, 238
FDCB05F6	253, 203, 5, 246
FDCB05FE	253, 203, 5, 254
FDE1	253, 225
FDE3	253, 227
FDE5	253, 229
FDE9	253, 233
FDF9	253, 249
FE20	254, 32
FF	255

Reader Service—see page 226

Livestock management via micros.

When the Cows Come Home



Sherill B. Nott
Dept. of Agricultural Economics
Michigan State University
East Lansing, MI 48824

Livestock farmers are now using large computer systems to store and process data on individual animals and herds. Regional interactive computers can be accessed over telephone lines for individual animal feeding recommendations. Dairy and goat farmers can enroll in the Dairy Herd Improvement Association (DHIA) to get monthly reports on milk production, as well as maintain permanent cow histories.

Dairy researchers have developed a herd reproductive management concept for which a microcomputer is ideal. A computer program can forecast a sequence of caretaker activities for breeding cows. The program can recommend whether or not to breed and which bull to use; it can schedule vaccinations and medical exams, and describe treatment for different health problems.

Computer Tasks

Four questions need answering when developing a breeding and health management system.

First, what management reports are needed? You can include a number of useful reports, including the following:

1. Calvings expected within the next X days.
2. Uterus exam needed because of recent calving.
3. No heats reported X days or more since calving.
4. Heats to be expected today.
5. Not pregnant for X days or more.
6. Pregnancy check needed today.
7. Dry off within the next X days.
8. Reproductive tract problems within the last X days.
9. Health treatments (not reproductive) made within the last X days.
10. Antibiotic treatments made within the last X days.
11. Vaccinations needed today.

Second, what information must be stored to get those reports? Several items must be filed regularly to insure timely and complete reports. The input options include:

1. Add a new animal to the disk.
2. Calving information.
3. Results of uterus exam.
4. Heats seen, receive bull to use.
5. Whether or not to breed and which bull to use.
6. Inseminations completed.
7. Results of pregnancy exam.
8. Cows dried off.
9. Reproductive health problems and treatments made.
10. Health (not reproductive) problems and treatments made.
11. Vaccinations completed.
12. Delete an animal from the disk.

Dairy managers may want to know everything about an animal or a subject.

The suggested information groupings are as follows:

13. Animal location, status, sex.
14. Vaccinations.
15. Calvings.
16. Pregnancy status, heats and breedings since last calving.
17. Reproductive health items.
18. Health (not reproductive) items.

Third, how will each record be packed? This project uses version 2.2 of TRSDOS, which packs each buffer (and hence each 256-byte disk record space) according to this format:

1 byte = alphanumeric (string) character.

2 bytes = integer

4 bytes = single-precision number.

8 bytes = double-precision number.

To store as much as possible, devise a set of codes to represent alphabetic information. Although string characters might seem to be the most effective, the possible code list is short. It includes 0 through 9, the alphabet and a few other symbols—well under 100 codes. Integers use two bytes but the possible codes range from −32,768 to +32,767; over 65,000 definitions can be set. Floating point numbers can be stored directly as data or can be treated as codes.

Fourth, will code numbers or alphanumeric data be stored?

A dairy farmer who knows cows by name should get to know them by number. If he has a choice between identifying a cow as Penelope or 9998, the trade-off is clear. 9998 requires two bytes; Penelope needs eight. Integers use less disk space, and most farmers al-

the electric pencil II™

for the TRS-80 Model II* Computer

The Electric Pencil is a Character Oriented Word Processing System. This means that text is entered as a continuous string of characters and is manipulated as such. This allows the user enormous freedom and ease in the movement and handling of text. Since lines are not delineated, any number of characters, words, lines or paragraphs may be inserted or deleted anywhere in the text. The entirety of the text shifts and opens up or closes as needed in full view of the user. Carriage returns as well as word hyphenation are not required since each line of text is formatted automatically.

As text is typed and the end of a screen line is reached, a partially completed word is shifted to the beginning of the following line. Whenever text is inserted or deleted, existing text is pushed down or pulled up in a wrap around fashion. Everything appears on the video display screen as it occurs thereby eliminating any guesswork. Text may be reviewed at will by variable speed or page-at-a-time scrolling both in the forward and reverse directions. By using the search or the search and replace function, any string of characters may be located and/or replaced with any other string of characters as desired. Specific sets of characters within encoded strings may also be located.

When text is printed, The Electric Pencil automatically inserts carriage returns where they are needed. Numerous combinations of Line Length, Page Length, Character Spacing, Line Spacing and Page Spacing allow for any form to be handled. Right justification gives right-hand margins that are even. Pages may be numbered as well as titled.

the electric pencil

—a Proven Word Processing System

The TRSDOS versions of The Electric Pencil II are our best ever! You can now type as fast as you like without losing any characters. New TRSDOS features include word left, word right, word delete, bottom of page numbering as well as extended cursor controls for greater user flexibility. BASIC files may also be written and simply edited without additional software.

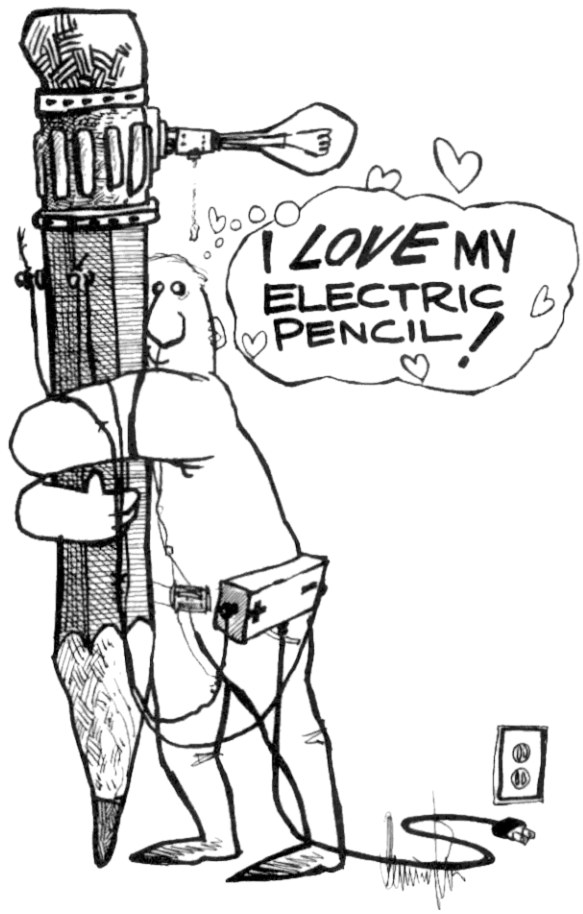
Our CP/M versions are the same as we have been distributing for several years and allow the CP/M user to edit CP/M files with the addition of our CONVERT utility for an additional $35.00. CONVERT is not required if only quick and easy word processing is required. A keyboard buffer permits fast typing without character loss.

	CP/M	TRSDOS
Serial Diablo, NEC, Qume	$ 300.00	$ 350.00
All other printers	$ 275.00	$ 325.00

The Electric Pencil I is still available for TRS-80 Model I users. Although not as sophisticated as Electric Pencil II, it is still an extremely easy to use and powerful word processing system. The software has been designed to be used with both Level I (16K system) and Level II models of the TRS-80. Two versions, one for use with cassette, and one for use with disk, are available on cassette. The TRS-80 disk version is easily transferred to disk and is fully interactive with the READ, WRITE, DIR, and KILL routines of TRSDOS.

TRC	Cassette	$ 100.00
TRD	Disk	$ 150.00

✓255



Features

TRSDOS or CP/M Compatible * Supports Four Disk Drives * Dynamic Print Formatting * Diablo, NEC & Qume Print Packages * Multi-Column Printing * Print Value Chaining * Page-at-a-time Scrolling * Bidirectional Multispeed Scrolling * Subsystem with Print Value Scoreboard * Automatic Word & Record Number Tally * Global Search & Replace * Full Margin Control * End of Page Control * Non Printing Text Commenting * Line & Paragraph Indentation * Centering * Underlining * Boldface

VISA

master charge THE INTERBANK CARD

*TRS-80 is a registered trade mark of Radio Shack, a division of Tandy Corp.

mss

MICHAEL SHRAYER SOFTWARE, INC.
1198 Los Robles Dr.
Palm Springs, CA. 92262
(714) 323-1400

✓Reader Service—see page 226

(from the creators of)
The Pencil Sharpeners™ & The Star Brightener™

now comes

THE POST *Script*™

POWERFUL MERGE UTILITY

(personalized form letters with data from your files)

and

IMPRESSIVE PRINT PACKAGE

(full word processing features of the major systems)

ENHANCEMENTS FOR

ELECTRIC PENCIL OR SCRIPSIT

Software Services ✓484

18325 Vanowen #34

Reseda, CA 91335

(213) 705-5999

COMPUTER PAPER
AND LABELS

9½ x 11
(8½x11 Detached)
$21.95

1 PART BLANK
3,000 to Box
(F.O.B. St. Louis MO.)

14 7/8 x 11
(Same Detached)
$26.95

Pressure Sensitive Labels

15/16 x 3½

1 Across.....$9.95/5M — 2 Across.....$19.95/10M

VISA

master charge

655 MESCALERO **FORM VILLAGE** ✓293 St. Louis MO. 63031

TOLL FREE 800 325-8987 IN MISSOURI: 314-831-3535

TRS-80® HELP ?

"...how did I get along without it?" is the usual reaction to this advanced disk indexing program.

AUTOMATICALLY create, sort, print, search a Master Index of all disk files.
AUTOMATICALLY read file names, disk numbers *(no hand entry)*.
AUTOMATICALLY print disk labels *(optional)*.
AUTOMATICALLY update Index from revised disks.
AUTOMATICALLY add optional file descriptions.
AUTOMATICALLY label disks as 'DOS' or 'DATA'. **NEW**
AUTOMATICALLY purge disks of unwanted files. **NEW**

- **FINDISK-II** *(on disk or tape 32K min — now works with RS 1.1 c mod)* $20.00

Other powerful programs from Documan available on tape or disk:

- **STRUCT-I** beam design and moment transfer w/graphics *(16K)* $15.00
- **SOLAR-I** essential calculations for passive solar design *(32K)* $30.00
- **RIA-I** complex analysis of real estate investments *(32K)* $30.00
- **DEPRECIATE-I** calculate, print 12 facts on depreciable items (32K) $15.00

✓88 **VISA/MASTER CHARGE** Shipping $1.00 each (Mich. add sales tax)

DOCUMAN SOFTWARE BOX 387-A KALAMAZOO, MI 49005 (616) 344-0805

now you can

RAISE

and tilt your
more

"80-LNW" VERSION NOW AVAILABLE TO SHROUD THE LNW INTERFACE BOARD

$25.50+$2.00 Shipping & Handling

Silver-gray
erglass module
fits under the
"80" monitor.

$22.50 + $2.00 shipping and handling with check or money order. NYS residents add 7% tax.

SYRACUSE R&D CENTER
Box 125, DeWitt, N.Y. 13214
"our 10th year in R&D" ✓358

TUTIL Examine, clear, initialize, move and modify data in memory. Compare two blocks or search for up to 24 byte HEX or ASCII string. Punch, load, verify, or execute SYSTEM tapes. Set breakpoints and jump. Display and modify registers. Output to printer if desired. TUTIL tape has a bonus program. MODIFY improves EDTASM(1.1 or 1.2) by allowing return to BASIC without loss of source file. Also uses Level II I/O to allow debounce, serial drivers, etc. (16-48K) $15
DUTIL Has all the above plus disk read and write. (16-48K DOS) $20

DISK-MOD Modify EDTASM(1.1 or 1.2) under TRSDOS, NEWDOS, even VTOS into a disk type E/A. Adds block move, global change, paged output to printer with optional page prompt, sorted symbol table, and DEFM corrections. Protect memory, use Level II I/O block, recover from reboots, and branch to any address. DIR, FREE, and KILL available without leaving the modified E/A (32F+ DOS required). $20

DISMBLE I Disassemble Z80 code to video, printer or tape. Provides EQUates ORG, and even labels. Tapes load into E/A to allow you to modify machine code programs easily. (16K, 32K, and 48K) $15
DISMBLE II DOS version writes E/A or MACRO disk file. (32K DOS) $20

STEP80 will single step BASIC or machine language programs even in ROM. Outputs trace values to video or printer via DCB. STEP80 in ROM. Outputs trace values to video or printer via DCB. STEP80 more. Relocateable program. $16.95 + $1.00 postage and handling.

SILVER-IT a kit to silver solder the TRS80's cheap bus connections for increased reliability and wear. Hi-Mo silver away for only $5.

INSIDE LEVEL II excellent manual containing 18 chapters on using ROM routines with many examples and explanations. It shows how to make a composite program load under SYSTEM but execute in both BASIC and machine language. Problems of using ROM in a disk system are shown and solved. Expand USR; relocate and interface with BASIC programs after reading these chapters. $15.95 + $1.00 postage and handling.

THE BOOK The most comprehensive and complete book yet on the math routines in Level II ROM. Has a commented disassembly of addresses 70H to 1607. This is the BOOK for the serious assembly programmer. 136 pages. 7 chapters. $14.95 plus $1.50 postage and handling.

DOS Bytes BOOT, SYS0/SYS, SYS1/SYS comments. 12 page booklet. $5

✓102

TRS-80* SOFTWARE! THE BEST

THE DATA ORGANIZER

*Variable length records
64 K Mod II $250.00

*Max. 20 field per record
32 K Mod I $150.00

MAGAZINE DISTRIBUTION PROGRAM

- 500 titles
- Invoices
- 200 Vendors
- Reports

64 K Mod II $750.00

AMBULANCE BILLING SYSTEM

- 1000 Calls/month
- Reports unpaid, medicare
- Tracks cars/drivers

64 K Mod II $750.00

DEALER OF TARANTO & ASSOCIATES MODEL II SOFTWARE
A/R G/L P/R each $249.00
All programs error free and fully documented. User tested.
Client list available. Custom programming/consulting for TRS-80*.

CMS, INC. 3132 N. BROADWAY, CHICAGO, IL 60657
(312) 327-7550 ✓393

*A trademark of Tandy Corp.

ready accept code numbers. Integers ease the task of writing software because they can be used directly as array indexes. A suggested set of descriptions with codes and variables for herd use is in Table 4.

Dairy Cow Files

The following file formats will meet your reporting needs and will work as random access files within the constraints of the TRSDOS. I offer the formats, records and variables as a starting point for your own project.

Discuss them with farmers, ranchers, animal scientists, veterinarians and other livestock professionals. The final decisions are too important to be left only to systems analysts and programmers!

I set up three random access records of data for each cow. On TRSDOS diskettes 335 records are available. Keeping all three records for a cow on one disk, 100 cows use 300 records. This leaves 35 records for other uses.

I use records 305 and 310 as indexes to keep track of which cow identification numbers are at which record locations.

In accessing the disk, the computer first gets record 310 and finds the cow's record number. If the cow is not on the disk, an appropriate message is printed. If the cow is on the disk, the variable JCOW is set equal to the cow's record number. The first record carrying status, calvings and breeding can be put or retrieved at JCOW, the second at JCOW + 100 and the third at JCOW + 200.

Tables 1, 2 and 3 give the detailed descriptions of space utilization, buffer variables and random access memory variables for records 1, 2 and 3, respectively.

Although each record can have 256 bytes, none of the suggested formats use it all. Some space is left for future data additions. Also, the machine will run better if it isn't filled to capacity.

To store RAM variables on a disk, you must reset the RAM variables into fielded buffer variables and put the buffer variables onto the disk. To get data from a disk, the computer takes one record and places it in a fielded buffer. The buffer values are then reset into RAM variables for manipulation. Several buffers may be activated simultaneously, thus making several records of data available.

Each record starts with the cow identification number and a

	Space on Disk	Buffer Variable BS(119)	RAM Variables KS(15) KC( ) KB( )
Animal ID No.	2	1	1
Disk Location No.	2	2	2
Date of Antibiotic Use	2	3	3
Birth date	2	4	4
Pen No.	2	5	5
Vaccinations 5 codes	10	6 : 10	6 : 10
Sex and Calving Status	2	11	11
No. of Rows Used—Calvings	2	12	12
8 Calving Sets or Rows:			KC(8,8)
Calving date	2	13	1 to 11: 1
Calf sex	2	14	2
Calf status	2	:	3
Cow status 1	2	:	4
2	2		5
3	2		6
4	2	75	7
Calf's sire	2	76	8
Breeding Availability	2	77	13
Bull to use for Breeding	2	78	14
No. of rows used—heats, bred	2	79	15
8 Sets or Rows of Heats, breeders:			KB(8,5)
Heat Date	2	80	1 to 11: 1
Who saw heat	2	81	2
Breeding date	2	.	3
Bull used	2	118	4
Inseminator	2	119	5
	238 spaces used		
	256 spaces available		

Table 1. Cow Status, Calvings and Breedings

✓Reader Service—see page 226

	Space on Disk	Buffer Variable BR(55)	RAM Variables LR(6) LH( ) L$( )
Animal ID No.	2	1	1
Disk Location No.	2	2	2
Placenta Status Code	2	3	3
Vaginal Discharge Code	2	4	4
Tract Status at Insemination	2	5	5
No. of Rows Used Below	2	6	6
11 Observation sets or rows as follows:			LH(11,3) I = 1 to 11
Date of Problem	2	7	J = 1 = date J = 2 = problem J = 3 = route
Health Problem	2	8	L$(11,3) I = 1 to 11
Treatment route	2	49	J = 1 = Calendar Date J = 2 = Treatment made
Treatment Made	15	50	
	243 spaces used 256 spaces available		

Table 2. Reproductive Tract Health Items

record (JCOW) number. These two variables let you check for errors. If either number brought into RAM is not what you expect, an error message is printed and processing is stopped until the problem is fixed. If parts of the disk become unreadable, the two variables may help in retrieving what is readable.

Dates are stored repeatedly in all three records. These are not calendar dates, but numbers relative to a fixed day. (See Listing 1.) January 1, 1968 is my constant date 1. Date 366 is December 31, 1968 (a leap year), and date 367 is January 1, 1969.

A date constant lets you store a date as an integer. It also lets you find out how many days have elapsed since a specific task was performed. Date constants ignore the number of days in a month.

Software can make these transformations invisible; the farmer will input calendar dates and receive results in the usual month/day/year format.

Note in Table 1 that every entry on the disk is a two-byte integer. First come the animal and record (disk location) numbers for error checking. Next are the animal status codes showing the last date of antibiotic use, birthdate of the cow, and pen or corral number of her current location. Next is a set of codes for five different vaccinations.

Sex and calving status (see Table 4 for my definitions) will speed up generating reports where the complete disk is searched. The next value will range from 0 through 8, indicating how many calvings are in the record for the cow. Each calving set has eight codes: the date, calf sex, calf status at birth (normal, large, etc.), four cow status codes at time of giving birth (normal, milk fever, etc.) and the calf's sire.

If artificial insemination is used, the two codes on breeding availability and bull to use can be preset. The employee can then receive instructions on what action to take if the cow is in estrus.

The next value will range from 0 through 8, indicating how many heats and breedings are in the record. Each breeding set has five variables: heat date, who saw the heat, date bred, semen used and who did the inseminating. When a calving is reported for the cow, this last group of data is zeroed, because only data since the last calving is important.

Table 2 describes the second record. The animal and record (disk location) numbers are for error checking. Next come three codes for flagging cows that need special observation or treatments. The code definitions are in Table 4.

The placenta code is for the status at time of calving, and the *vaginal discharge code* is for following through after calving to detect metritis. The tract status at insemination flags any

abnormalities at time of breeding.

The next value will range from 0 through 11 and indicate how many reproductive observation sets are in the record. Each set has four variables: date observed, the problem (e.g., metritis, not ovulating), the route of treatment (e.g., intravenous, intramuscular), and a 15-place alphanumeric string to describe the treatment.

Forget trying to encode this with integers. Let the farmer type on the keyboard what seems best. Although this will use more disk space, the task of creating codes for all possible treatments found at the farm is doomed to failure.

You may find that 15 characters are too few. Another alternative is 24 characters to the treatment string and eight instead of 11 stored sets. Remember the 256-byte constraint.

Table 3 describes record 3. The animal and record (disk location) numbers are for error checking. The next value will range from 0 to 11, indicating how many health sets are in the record. Each set has four variables; date observed, problem code (a short list can suffice if "other" is included), the route of treatment (oral, intramammary, etc.) and a 15-place alphanumeric string for comments.

Demonstration Software

The hardware includes a TRS-80 Model I with 48K of RAM, a printer and two disk drives. The first drive stores the software; the second drive stores only animal data. Version 2.2 of TRSDOS and disk BASIC are used. A structured programming approach results in several short subroutines.

The following previously defined reports and tasks were programmed and debugged: not pregnant for X days or more; add a new animal; calving information; health (not reproductive) problems and treatments; delete an animal; calvings; and health items (not reproductive).

These will demonstrate the system.

Much effort went into building what I call overhead subroutines and standardized arrays. Each of the three records need to be opened, fielded, read from disk to buffer and reset into RAM variables. Then the steps need to be reversed to store data on the disk. Individual subroutines do these jobs.

If more than 100 animals are to be recorded, another subroutine waits for a different disk to be loaded into drive 2. The two date-transformation subroutines are overhead.

All the arrays are defined and dimensioned in the main program. Management action report 5, not pregnant for X days or more, was the only one activated. To activate the next report requires only a few subroutine calls, some manipulation of already-defined variables and a few print statements.

The software has 632 lines, including remarks. Using the PRINT MEM command before and after loading the program into RAM indicates the program size is almost 24K.

After running the program to dump one record for a cow, PRINT MEM indicates the program plus execution space required (26.9K). This leaves only 10.7K of space to add options to the program. Given the hardware and software used, all options cannot be activated in one big program.

Some Variables

A complete dairy livestock management system should include milk weights. Each milk weight will be a decimal number using four bytes of space. Two observations per day for 31 days requires 248 bytes, nearly a record. Part of a second record is needed on the few farms that milk three times per day.

If you try to integrate milk weights with breeding, calving

	Space on Disk	Buffer Variable BH(50)	RAM Variables
Animal ID No.	2	1	I1
Disk Location No.	2	2	I2
No. of Rows Used Below	2	3	NG(4)
11 observation sets or rows as follows:			IH(11,3)
Date of Problem	2	4	I = 1 to 11 J = 1 = Problem J = 2 = Route J = 3 = Constant Date
Health Problem	2	5	B3(11,2)
		:	I = 1 to 11
Treatment Route	2	46	J = 1 = calendar date string
Treatment Made	15	47	J = 2 = treatment made string
	239 spaces used of 256 available		

Table 3. Health Items (Not Reproductive)

Variable	Code	Definition
Calf sex:		
T1$(0)	0	Unknown
T1$(1)	1	Female
T1$(2)	2	Male
T1$(3)	3	Female twins
T1$(4)	4	Male twins
T1$(5)	5	Mixed twins
Calf status at birth:		
T2$(0)	0	Unknown
T2$(1)	1	All normal
T2$(2)	2	Born dead
T2$(3)	3	Found dead
T2$(4)	4	Alive, crippled
T2$(5)	5	Large calf
Cow status at time of giving birth:		
T3$(0)	0	Normal, no comments
T3$(1)	1	Assisted delivery
T3$(2)	2	Aborted or stillborn
T3$(3)	3	Milk fever
T3$(4)	4	Udder broken down
T3$(5)	5	Weak back legs
T3$(6)	6	Downer cow
T3$(7)	7	Large calf
T3$(8)	8	Retained placenta
T3$(9)	9	Cleaned by hand
Animal sex and status code:		
T4$(0)	0	Unknown or other
T4$(1)	1	Female, never had a calf
T4$(2)	2	Female, calved at least once, now in milk
T4$(3)	3	Female, calved at least once, now dry
T4$(4)	4	Female, in feedlot
T4$(5)	5	Male, castrated
T4$(6)	6	Male, not castrated
T4$(7)	7	Male for breeding
Route of giving health treatments:		
T5$(0)	0	Unknown or other
T5$(1)	1	Intermammary
T5$(2)	2	Intermuscular
T5$(3)	3	Interperitoneal
T5$(4)	4	Intervenous
T5$(5)	5	Oral
T6$(6)	6	Ulterous infusion
Reproductive health problems:		
P1$(0)	0	Unknown
P1$(1)	1	Retained placenta
P1$(2)	2	Post calving discharge
P1$(3)	3	Metritis
P1$(4)	4	Pus on catheter
P1$(5)	5	Sticky when bred
P1$(6)	6	Scar tissue
P1$(7)	7	Other abnormality
Health problems (not reproductive):		
P2$(0)	0	Unknown
P2$(1)	1	Mastitis
P2$(2)	2	Footrot
P2$(3)	3	Milk fever
P2$(4)	4	Off feed
P2$(5)	5	Diarrhea
P2$(6)	6	Pneumonia
P2$(7)	7	Calving problems
P2$(8)	8	Hardware
P2$(9)	9	Other problems
Breeding status code:		
T6$(0)	0	Unknown
T6$(1)	1	Open, breed if other criteria are met
T6$(2)	2	Assumed to be pregnant
T6$(3)	3	Palpated, known to be pregnant
T6$(4)	4	No, do not breed
T6$(5)	5	Running with a bull
T6$(6)	6	Not relevant

Table 4. Suggested Variables, Code Numbers and Definitions

and health data on a single diskette, the result may have to be quite different from Tables 1, 2 and 3.

The formats I've discussed are designed for breeding livestock using artificial insemination. If natural service is used, much of the herd reproductive approach becomes irrelevant. Most of record 1 and all of record 2 is not needed.

In feeder operations (beef, hogs, lambs) the file formats should be devoted more to preventive health strategies, daily feed amounts and weight gains. As in the dairy example, disk space is the first problem to consider.

A different approach would involve a series of separate programs, one for input and another for generating reports or editing. Redundant program lines make writing the second and third programs easier.

One disadvantage to this is that the farm worker has to both load and run a series of different programs during each session and shuffle the data disks. But TRSDOS offers a chain option for automatically loading and running a different program.

In this project all data inputs are taken from the keyboard after a screen prompt. As soon as the input is read it is checked for errors. I devoted considerable attention to idiot-proofing. This is an adequate input technique given the speed with which the computer operates when moving among RAM, the screen and the keyboard.

Study Other Systems

Several research herds have had computerized data storage and inventory programs created for them. The package developed for the U.S. Dept. of Agriculture's research herd at Beltsville, MD, is mentioned in the *Journal of Dairy Science* (May 1974, p. 611). The Michigan State University dairy research center at East Lansing uses another system.

Both systems can be studied for data handling procedures and coding schemes. The software accounts for many research functions and institutional inventory requirements not needed in a privately owned herd.

The herd reproductive management concept has been implemented on several computers. A mail-in batch-operated software package written in PL-1 was created and field tested on several dairy farms by Virginia Polytechnic Institute and State University.

I set up software in FORTRAN on an interactive time-share computer operated by the Michigan State University Cooperative Extension Service. It was tested on only two herds, and remains available on a trial basis.

At least one private firm, Herd Reproductive Services, Inc., Athens, GA, has developed software for managing herd reproduction. This firm will provide a farmer with a minicomputer, the software and monthly consulting services.

Opportunities exist for creating management-oriented livestock breeding and health record systems on microcomputers. However, the planning process will take time and demand a thorough knowledge of livestock. ■

References

Beall, Gary, "Good Records Don't Have to be Complicated," *Hoard's Dairyman*, February, 1976.

Hughes, Joan K., and Jay I. Michtom, *A Structured Approach to Programming*, Prentice-Hall, Inc., 1977.

Miller, R.H.; M.E. Creegan; and R.E. Pearson, "Computer Systems for Herd Reproduction and Health Data," abstract of paper 38, *Journal of Dairy Science*, May, 1974.

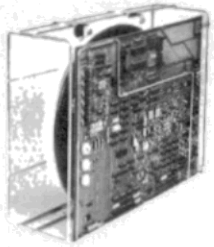
```
** CALVING INFORMATION STORED FOR COW ID =  50
CURRENT STATUS = FEMALE, CALVED AT LAST ONCE, NOW IN MILK
LN BIRTH DATE  CALF SEX     CALF AT BIRTH    COW AT BIRTH        BULL
 1  1/ 1/1979  FEMALE        ALL NORMAL      NO COMMENTS 0
 2 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 3 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 4 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 5 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 6 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 7 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
 8 UNKNOWN     UNKNOWN       UNKNOWN         NO COMMENTS 0
```

Sample Run 1. Calving information stored for Cow ID = 50.

```
** TO WRITE OUT HEALTH RECORDS **
HEALTH INFORMATION ON FILE FOR ID  45
LN    DATE        PROBLEM          TREATMENTS
 1  1/11/1979    MASTITIS    INTRAMAMMARY   2 TUBES ANTI-BI
 2  1/ 1/1979    MASTITIS    INTRAMAMMARY   3 TUBES OF ANTI
 3 11/15/1979    MASTITIS    INTRAMAMMARY   4 TUBES ANIT-BI
 4 11/16/1979    MASTITIS    INTRAMAMMARY   4 TUBES ANTI-BI
 5 11/17/1979    MASTITIS    INTRAMUSCULAR  PENICILLIN
 6 11/19/1979    OFF - FEED  ORAL           MOLASSES
 7 EMPTY         UNKNOWN     UNKNOWN        EMPTY
 8 EMPTY         UNKNOWN     UNKNOWN        EMPTY
 9 EMPTY         UNKNOWN     UNKNOWN        EMPTY
10 EMPTY         UNKNOWN     UNKNOWN        EMPTY
11 EMPTY         UNKNOWN     UNKNOWN        EMPTY
```

Sample Run 2. Health information on file for ID 45.

PRIAM
Hard Disks Now Available from SIRIUS SYSTEMS!



PRIAM's high-performance, low-cost Winchester disc drives speed up throughput and expand data storage from 20 megabytes to 154 megabytes. And a single controller can be used to operate 14-inch-disc drives with capacities of 33, 66, or 154 megabytes or floppy-disc-size drives holding 20 and 34 megabytes. So it's easy to move up in capacity, or reduce package size, without changing important system elements or performance.

- *Fast, Linear Voice Coil Positioning*
- *10 ms track-to-track positioning*
- *Fully servoed head positioning*
- *Dedicated servo tracks*
- *DC Power required only!*
- *Simple, parallel Interface*
- *Optional SMD Interface*
- *50 ms Average Positioning time*
- *90 ms Maximum Positioning Time*
- *6.4 ms Average Latency*

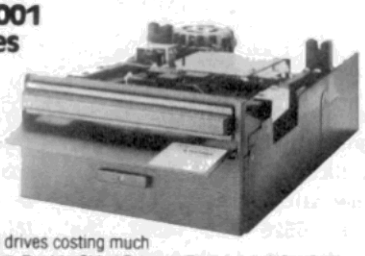
THE PRIAM LINEUP

Model/Disc Size	Capacity	Size	Weight	Price
DISKOS 3350 (14")	33Mbytes	7" × 17" × 20"	33 lbs.	$2995
DISKOS 6650 (14")	66 Mbytes	7" × 17" × 20"	33 lbs.	$3749
DISKOS 15450 (14")	154 Mbytes	7" × 17" × 20"	33 lbs.	$4695
DISKOS 2050 (8")	20 Mbytes	4.62" × 8.55" × 14.25"	20 lbs.	$2995
DISKOS 3450 (8")	34 Mbytes	4.62" × 8.55" × 14.25"	20 lbs.	$3745
DISKOS 570	5.3 Mbytes	floppy-size	(low)	(low)
DISKOS 1070	10.6 Mbytes	floppy-size	(low)	(low)

All PRIAM DISKOS Drives have a Transfer Rate of 1.03 Mbytes/Sec.
Optional SMD interface available for $150.

SIRIUS SYTEMS offer cases and enclosures for all PRIAM Hard Disk Drives. All 14" Winchester Drives will mount in our 14" Standard Case. The 8" Winchesters have two alternatives: a single drive case and a dual drive case. All SIRIUS SYSTEMS Winchester drive cases include Power Supply, internal cabling, switches, fan, extra AC outlet (not switched, but fused) and possess very adequate ventilation. Drive addressing is done on the rear of the Case and not on the drive itself to provide ease of use during operation. All WINCHESTER DRIVE Cases are Warranted for a full year and come in our standard blue-black color scheme. Consult us for current availability and pricing.

Remex RFD 4000/4001 8" Floppy Disc Drives Double sided . . . Double density!!



$549.95

RFD 4001, $569.95

Offers quality and features found in drives costing much more! ■ Single or Double Density ■ Double-Sided Drive ■ Door Lock INCLUDED ■ Write-Protect INCLUDED ■ 180 Day Warranty ■ Compatible with Shugart 850/851 ■ Low Power Operation ensures LONGER LIFE!! ■ Model RFD 4001 offers Data and Sector Separator

RFD 4000/4001 Technical Manual	6.95		
Connector Set #3 (AC, DC, Card Edge)	10.95	RFD 4000C/B Cabinet (for use with	
Connector Set #4 (AC and DC)	2.95	Power Modules)	29.95

Remex 1000B . . . If you've been looking for a less expensive floppy disc drive, but not wanting to sacrifice quality — this is it!

$419.95

You get both in the Remex 1000B! For only $419.95 look at what you get: ■ 8" Floppy Drive ■ Single or Double Density ■ Hard or Soft Sectoring ■ Media Protection Feature ■ Single Density Data Separator ■ 180 Day Factory Warranty

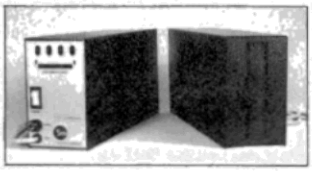
Door Lock Option	$19.95	Write Protect Option	$19.95	RFD 1000B Technical Manual	$5.95
Interface Adapter (REMEX-to-Shugart)	$14.95	Connector Set #1 (AC, DC, & Card Edge)	$10.95	RFD 1000B CASE (for use with Power Modules)	$29.95

SIRIUS 8" DISK POWER MODULES

The Single and Dual Drive Power Modules are designed to provide DC and (switched) AC power for one (the Single Drive Power Module) or two (the Dual Drive Power Module—the DDPM will power three RFD 4000s or 4001s) 8" Floppy Disk Drives. Many features are included for safe and reliable operation and the Power Modules come with our standard 180 day WARRANTY (the Open Frame Power Supply warranty is for 2 years). All Power Modules will work with either the RFD 4000C/B or RFD 1000B case (color schemes match also).

Dual Drive Power Module (DDPM)	$139.95
Single Drive Power Module (SDPM)	119.95

SIRIUS 80+
Perfect Add-Ons for Your Computer System!



The SIRIUS SYSTEMS 80+ Series of Floppy Disk add-ons are designed to provide unmatched versatility and performance for your computer. Consisting of four different add-ons, there is a 80+ Series Floppy Disk to meet your need. All 80+ Series Floppy Disk are compatible with the TRS-80* and come ready to plug in!

COMMON CHARACTERISTICS

- *5 ms track-to-track access time*
- *Auto-eject*
- *180 day WARRANTY*
- *Exceptional speed stability — 1½%*
- *Single density (FM) or double density (MFM/M²FM)*
- *Ultra high reliability*
- *2 year Power Supply Warranty*
- *Mix any or all 80+ Series on the same cable!*
- *Includes user accessible plugboard for drive reconfigurng*

SPECIFIC CHARACTERISTICS

The **SIRIUS 80+1** is a single sided, 40 track, highly reliable Floppy Disk add-on. Offering 5 more tracks than the Radio Shack model, it cost $140 less! Formatted data storage is 102K/20K bytes single/double density.

SIRIUS 80+1 ..... **$359.95**

The **SIRIUS 80+2** is a dual sided, 70 track (35 per side), highly versatile Floppy Disk unit. It appears to the TRS-80* as TWO 35 track drives, yet COST LESS THAN HALF THE PRICE! Even greater savings result, since data is recorded on both sides of the media instead of only a single side. Using the plug board, it may be reconfigured for other computer systems! (The 80+2 operates as Drive 0 and any of the other three addresses (with the sandard Radio Shack Cable) or as any of four drives (with the SS Standad Cable).) Formatted data storage is 80.6K/161.2K bytes single/double density.

SIRIUS 80+2 ..... **$449.95**

The **SIRIUS 80+3** is a single sided, 80 track, "Quad" density Floppy Disk unit. Offering 2½ times the storage of a Standard Radio Shack drive, the 80+3 greatly reduces the need for diskettes correspondingly. Additionally, because of the increased storage and faster track-to-track access time, the 80+3 allows tremendously increased throughput for disk based programs!!! The 80+3 INCLUDES SIRIUS's TRAKS-PATCH on Diskette. Formatted data storage is 204K/40K8 bytes single/double density.

SIRIUS 80+3 ..... **$489.95**

The **SIRIUS 80+4** Floppy Disk add-on is a double sided, 160 track (80 per side), 5¼" monster! The ultimate in state-of-the-art 5¼" Floppy Disk technology, to 80+4 is seen by the TRS-80* as two single sided disk drives, each with 80 tracks. Thus, in terms of capacity one 80+4 is equivalent to 4⅔ standard Radio Shack drives — a savings of over 73% (not to mention diskettes!!!). (With a double density converter, the available memory is huge!) The 80+4 is similar to the 80+2 in that it arrives configured as Drive 0 and any of the other three addresses (with the standard Radio Shack Cable) or as any of four drives (with the SS Standard Cable). The 80+4 INCLUDES TRAKS-PATCH on Diskette. (The plug board is also included.) Formatted data storage is 408K single density or 816K bytes double density.

SIRIUS 80+4 ..... **$624.95**

Ø

All 80+ Series Floppy Disk add-ons operate a 5 milliseconds track-to-track access time (eight times faster than the SA 400) but are Expansion Interface Limited to 12 milli-seconds for the TRS-80*.

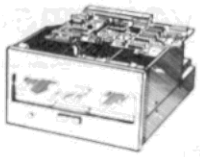
*TRS-80© Tandy Corp.

MPI 51/52 . . .
A Great Reliable Mini-Drive!

- *Fast! 5ms track to track access*
- *Exclusive Pulley-Band Design*
- *Unique Door/Ejector Mechanism*
- *Reliable 1½% Speed Stability*
- *Single/Double Density Operation*
- *Industry/ANSI Standard Interface*

MPI 51
(Single Head, 40 tracks, 120K/240K bytes Single/Double Density**) **$259.95**

MPI 52
(Dual Head, 70 tracks, (35/side), 218.8K/437.5K Single/Double Density**) **$349.95**



MPI 91/92 . . . NEW STATE-OF-THE-ART DISK DRIVE!

MPI 91
(Single Head, 80 tracks, 240K/480K Single/Double Density**) **$389.95**

MPI 92
(Single Head, 160 tracks (80/side), 480K/960K Single/Double Density**) **$499.95**

**Unformatted data storage*

Introducing the Versatile, Low-Cost OMEGA Series Controller

As new technological advances bring down the cost of fast, reliable mass data storage, the need for an inexpensive, versatile controller have become greater and greater. To meet this need, SIRIUS SYSTEMS' OMEGA Series Controller was designed.

The SIRIUS OMEGA Series Controller Module utilizes an **on-board microprocessor** to mediate data transfer to a wide variety of peripherals from an equally wide variety of host computer systems. Up to four Winchester Hard Disks (8" or 14"), four 5¼" Floppy Disks Drives and/or up to eight 8" Floppy Disk Drives may be **in use at one time.** Host systems interfacing is accomplished via a parallel or a serial interface. With the additon of a Personality module, the OMEGA Series Controller Module is directly **compatible with many popular computer systems** (among them the TRS-80*, Apple, Heath, and others). Provision is made for the addition of a streaming tape drive, also.

SPECIFIC HARDWARE FEATURES INCLUDE:

- *Control of up to twelve Floppy Disk Drives (eight 8" and/or four 5¼")*
  - *8" and/or 5¼" Disk Drive Utilization*
  - *Single (FM) or Double (MFM) density data storage*
  - *Hard or Soft sectored diskette usage*
  - *Utilization of "Quad" density (96 tpi) 8" or 5¼" Disk Drives*
- *Control of up to four WINCHESTER type PRIAM DISKOS Disk Drives*
  - *8" or 14" may intermix on the same cable*
  - *Accommodates 8" and/or 14" drives of 5.3Mbytes to 154Mbytes*
  - *Ultra-Fast data transfers*
- *Extremely flexible host-controller interfacing*

SPECIFIC SOFTWARE FEATURES INCLUDE:

- Dynamic format modifications via command words
- Extremely flexible format acceptance for unusual data storage formats
- Easily interfaces to standard operating systems (TRS-DOS*, CP/M***, etc)
- Operates in either get/put sector mode or data string mode
- Performance parameters may be changed by EPROM replacement or Dynaminic Reprogramming

Dedicated systems cards are also available on a limited basis for the STD-BUS and the S 100. These cards feature shared memory also (again, software selectable) in addition to the regular OMEGA Series Controller Module features. Consult SIRIUS SYTEMS for current price and availability for the entire line of OMEGA Series Memory Units and Controllers. **Dealer inquiries are invited.**

TO ORDER CALL (615) 693-6583

Phone Orders Accepted 9AM-7PM (ESDT)

We accept **MC, VISA, AE, COD** (requires Certified Check, Cashier's Check or Cash) and **Checks** (personal checks require 14 days to clear). **SHIPPING AND HANDLING:** $7.00 per Floppy Disk Drive or 80+ Module ■ 5% for other items (any excess will be refunded) ■ **Foreign Orders** add 10% for Shipping & Handling. Payment in U.S. currency ■ Tennessee residents add 6% Sales Tax ■ **VOLUME DISCOUNTS AVAILABLE**

SIRIUS SYSTEMS

7528 Oak Ridge Highway
Knoxville, Tennessee 37921 ✓67

Set your memory size from BASIC.

Memory Sizer

Jack Decker
1804 West 18th Street Lot 155
Sault Ste. Marie, MI 49783

In magazines such as this one, you often see hybrid programs. Written in BASIC, they use a machine language subroutine called by the USR(X) function during execution.

The author of the program usually tries to make the loading procedure more convenient by embedding the machine language code in DATA statements within the BASIC program.

Overwriting

These DATA statements are usually read by a FOR-NEXT loop that POKEs it into its proper location in memory. Already you can see a source of potential problems, since the code is POKEd without regard to whatever else may already be there.

If you have KBFIX or some other machine language routine loaded and the BASIC program overwrites it, tough luck—you have just lost your routine and maybe hung up your computer in the bargain.

What's that, you say? I forgot that I had to turn off the computer, power up again, and answer the MEMORY SIZE? question to protect some machine language memory.

Yes, that does destroy any machine programs residing in memory. Now, what was that MEMORY SIZE? Had it written down somewhere. Should have written it on the cassette label, I suppose. Suddenly this "convenient" program leaves something to be desired.

All of these troubles can be avoided. For instance, Listing 1 shows one unusual method that will work when the machine code to be used is less than 256 bytes long and is completely relocatable. For demonstration purposes I'll use a machine program that places ASCII codes corresponding to the contents of register pair HL in the upper right-hand corner of the video display.

The machine language routine is stored in string variable B$. VARPTR(B$) returns the address of the string's first three memory locations containing the length of the string, the least significant byte and the most significant byte. If you POKE these latter two bytes into locations 16526 and 16527, the USR(X) locations, then B$ will be treated as a machine language program when the USR function is called.

Note that if the VARPTR locations are greater than 32767, as they could be in a system with more than 16K of memory, you must subtract 65536 to keep the POKE command within allowable limits.

If B$ is the first string variable to be assigned in a program, it will stay in the same location. However, if you choose to assign other string variables first, and if the length of those variables will be changed during the run of the program, then you'll have to find B$ in memory before executing the USR call. You can do this by repeating the commands in line 30 just before making each USR call. (This could be done in a subroutine.)

Avoid Power Up

The above method works for most relocatable machine language routines. But you can locate your routine in high memory, and avoid powering up the computer to set the memory size. Let's examine what happens when you answer the MEMORY SIZE? question.

As far as I can tell, the main locations for the memory size are 16561 and 16562. When you type a number in response to the memory size prompt, the computer subtracts 2 and stores that number in 16561 and 16562. So if you ever forget what memory size you started with, you can type:

```
? PEEK (16561) + PEEK (16562)*256 + 2
```

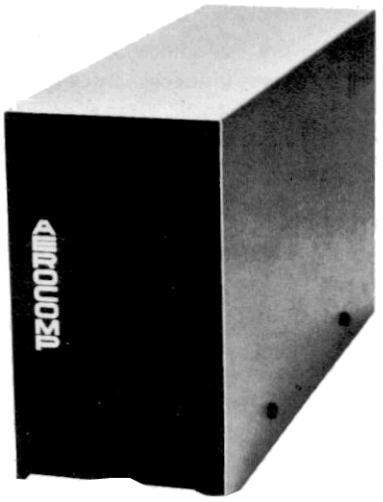
and the computer will tell you.

Actually, the address in those locations represents the highest usable memory location for BASIC, normally used for string storage. So, for example, if you set the memory size to 30000, the last location available for string storage (or other BASIC variables, if no string space is cleared) is 29998. You could start your machine program at 29999 with no ill effects.

But you can't just POKE the new value minus 2 into 16561 and 16562 to reset the memory size. Other pointers must also be reset—those that govern the

NEW

DISC DRIVES
FOR TRS—80*
"FLIPPY"—250K bytes/disc
FAST—5ms Track to Track Access



REAL VALUE

AEROCOMP offers the best value in microcomputer disc drives on the market today! Reliability, features and cost tough to beat. We deliver...and we stand behind our products, as evidenced by the only FREE TRIAL OFFER in the industry. Examine your systems needs and order today!

MYSTERY REMOVED

There appears to be some confusion in the terminology used to describe disc drives and their features. Here's what we mean:

•FLIPPY Allows the use of both sides of a diskette with a single-headed drive by simply turning the diskette over (Model 40-1).

•TRACK DENSITY specified in tracks per inch (TPI). Refers to the number of tracks per radial inch on the diskette. Typically 48 TPI=40 usable tracks and 96 TPI=80 useable tracks.

•DOUBLE DENSITY refers to recording density in bits per inch (bpi). Typically single density means data can be recorded up to 2,938 bpi; double density means data can be recorded up to 5,876 bpi.

•DOUBLE-SIDED refers to number of read/write heads. Single-sided is one head, read/write one side only; double-sided is dual heads allowing read/write operations on both sides of the diskette. A double sided drive appears as two separate drives to the controller.

•CAPACITY unformated capacity is the total amount of storage space available on a diskette. Typically 125K bytes on a 40 track 5.25in. diskette. Formated capacity is the total USABLE storage space on a diskette. Typically 102K bytes on a 40 track 5.25in. diskette.

•ACCESS TIME the time required for the head to move from one track to the next. Typically 5 to 40 milliseconds (ms).

FREE TRIAL OFFER

Order your AEROCOMP Disc Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return it, packed in the original shipping container, for a FULL REFUND. We have complete confidence in our products and we know you will be satisfied! ORDER TODAY!

WARRANTY

We offer you a 90 day unconditional warranty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes necessary, our service department is fast, friendly and cooperative.

100% TESTED

AEROCOMP Disc Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!

ORDER NOW

• MODEL 40-1 DISC DRIVE $339.95ea.
Single-sided, "Flippy", 48TPI.
(40 track; single density unformated 125K bytes/side; double density unformated 250K bytes/side).

• MODEL 80-2 DISC DRIVE $439.95ea.
Double-sided, 48TPI.
(80 track/40 per side; single density unformated 250K bytes; double density unformated 500K bytes).

• MODEL 80-1 DISC DRIVE $439.95ea.
Single-sided, "Flippy", 96TPI.
(80 track; single density unformated 250K bytes/side; double density unformated 500K bytes/side).

• MODEL 160-2 DISC DRIVE $595.95ea.
Double-sided, 96TPI.
(160 track/80 per side; single density unformated 500K bytes; double density unformated 1 megabyte).

All models are capable of single or double density and are complete with power supply and silver enclosure. Send for information on AEROCOMP 2- and 3-drive systems available in 40 and 80 track.

• SELECT EITHER A 2-DRIVE OR 4-DRIVE CABLE FOR USE WITH YOUR DRIVE(S):
2-DRIVE CABLE (for use with 1-or 2-drive systems) **$24.95ea.**
4-DRIVE CABLE (for use with 1-, 2-, 3-or 4-drive systems) **$34.95ea**
Add $1.25 shipping and handling

• MINI DISKETTES (5.25 in). Box of 10 **$29.95** add $1.25 shipping and handling

• DISC OPERATING SYSTEMS
NEWDOS+(40 TRACK) $109.00
NEWDOS/80 (80 TRACK) $149.00

★ SPECIAL COMBINATION OFFER ★

• Model 40-1 Disc Drive	$339.95
• 2 Drive Cable	24.95
• Disc Operating System (NEWDOS+)	109.00
• Freight	5.25
Reg. $479.15	
Special $399.95	

SPECIAL COMBO EXCEPT WITH NEWDOS/80
Reg. $519.15 **Special $429.95**

To order by mail, specify Model Number(s) of Drive, cable, etc. (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request COD shipment. Texas residents add 5% sales tax. ADD $4.00 per drive for shipping and handling. Please allow 2 weeks for personal checks to clear our bank. No personal checks will be accepted on COD shipments-cash, money orders or certifed checks only. You will receive a card showing the exact COD amount before your shipment arrives. Be sure to include your name and shipping address. WE SHIP PROMPTLY! In the event there is a slight delay, you will be notified of the shipping date and we will NOT deposit your money order or charge your bankcard until the day we ship!

COMPARE AND BUY AEROCOMP!

	"FLIPPY"	ACCESS TIME (track to track)	HEAD LOAD SOLENOID	DISC EJECTOR	CAPACITY (unformated single density)	EASY-ENTRY DOOR	FREE TRIAL
AEROCOMP	YES	5ms.	YES	YES	250K bytes (both sides)	YES	YES
RADIO SHACK*	NO	40ms.	YES	NO	109K bytes	NO	NO
PERCOM	YES	25ms.	YES	NO	250K bytes (both sides)	YES	NO
MPI	NO	5ms.	YES	YES	125K bytes	YES	NO
SHUGART	NO	40ms.	YES	NO	109K bytes	NO	NO
SIEMENS	NO	25ms.	YES	NO	125K bytes	YES	NO
TANDON	NO	5ms.	NO	NO	125K bytes	NO	NO
PERTEC	YES	25ms.	YES	NO	250K bytes (both sides)	NO	NO
BASF	NO	12ms.	YES	NO	125K bytes	NO	NO

Factual material from current manufacturer's data sheets is believed reliable but cannot be guaranteed, comparing Aerocomp Model 40-1 to similar models.

The TRS-80* expansion interface limits the track to track access time to 12ms. *Trademark of Tandy/Radio Shack.

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 24
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applications and Technical information, call (214) 337-4346 or drop us a card.

Dealers inqiries invited

AEROCOMP
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224 ✓387

✓Reader Service—see page 226

start of string storage and the position of variables in memory.

Fortunately, you don't have to reset each one individually. All you need to do is issue a CLEAR X command where X is the number of bytes you need cleared for the program. The X argument must not be omitted, even if you don't need any string space cleared, because that also resets a pointer. If you don't need any string space, you can just CLEAR 0.

Using the above information, if you wanted to reset the memory size to 30000, you could issue these commands:

```
POKE 16561,46 : POKE 16562,117 : CLEAR 50
```

46 + 117•256 = 29998, or 2 less than our desired memory size.

However, if your machine language routine is relocatable, you can go one better than this. You can reserve enough memory for your routine while the program is running, and then free that memory for other uses when the program is finished.

Freeing Memory

Take a look at Listing 2. The code is compressed with multiple statements on a line, so keep in mind that everything between lines 1 and 2 is part of line 1, and so on. Here is how the program works.

First, POKE 16396,23. This disables the break key since you don't want to exit the program without executing line 9999. You then PEEK at the present memory size in locations 16561 and 16562, and let variable C equal the present memory size minus the length (in bytes) of your machine routine. This will be your new memory size.

Poke your present memory size into 16526 and 16527. This is temporary since you want to recover these values after you issue the CLEAR command (which clears all variables). You could just as easily have used any of several other memory pairs to store these values, but these are the USR(X) locations that you will be using later anyway.

Next you break your new memory size down into least significant and most significant bytes (LSB and MSB), and POKE these back into 16561 and 16562. Then comes the mandatory CLEAR command (be sure to replace SC with a numeric argument).

Starting in line 2, you recover your original memory size from the USR(X) locations and store them in variables QY and QZ (to be used at the end of the program). Now we PEEK at your new memory size and add 1 to it to determine where your machine program will start (variable C).

If the entry point (the byte of the routine to be accessed first) is not the first byte of the routine, you must calculate the entry address by adding the displacement from the start of the program (variable D). Break this entry point address down into LSB and MSB and POKE it into the USR(X) locations (16526 and 16527). You then do the actual POKEing of your routine into memory, starting at the first location of your newly protected memory.

If the memory location indicated is greater than 32767, you must include the offset of −65536. This necessitates the function (A>32767)•65536 shown in the example. Note that PL indicates the program length and must be replaced by the number of bytes to be POKEd from DATA statements.

Restoring and Modifying

At the end of your program, line 9999 in this example, you need only rePOKE the old memory size into 16561 and 16562, issue the necessary CLEAR, and POKE 16396,201 to re-enable the break key. This restores everything to the way it was before you ran the program, without any need to power up again. Note that QY and QZ are the only variables that must be left unchanged throughout the program.

The above routines can also be modified to put a system machine code in your programs, that is, code that stays in memory and is used in the command mode.

Should you use one of the above routines, I would also suggest you use the CLEARX statement at the beginning of every program; you never know how much memory was reserved for strings in your last program. Finally, if your program really hogs the string space, issue a CLEAR 50 at program end so that you can load the next program without encountering an OM ERROR (out of memory error). ■

```
10 FOR A=1 TO 4: READ B: BS=BS+CHR$(B): NEXT
20 DATA 34, 62, 60, 201
30 B=VARPTR(BS): FOR A=0 TO 1: B=B+1: POKE 16526+A,PEEK
 (B+(B>32767)*65536): NEXT
40 CLS: A=USR(A)
```

Program Listing

```
0 REM

THE FOLLOWING CONSTANTS MUST BE INSERTED INTO PROPER PL
ACES:

PL (REPLACE WITH PROGRAM LENGTH - IN BYTES)
SC (REPLACE WITH STRING SPACE TO BE CLEARED - DO NOT OM
IT!!)
EP (REPLACE WITH ENTRY POINT - BYTE DISPLACEMENT FROM S
TART)

1 POKE16396,23:A=PEEK(16561):B=PEEK(16562):C=A+B*256-PL
:POKE16526,A:POKE16527,B:B=INT(C/256):A=C-B*256:POKE165
61,A:POKE16562,B:CLEARSC
2 QY=PEEK(16526):QZ=PEEK(16527):A=PEEK(16561):B=PEEK(16
562):C=A+B*256+1:D=C+EP:B=INT(D/256):A=D-B*256:POKE1652
6,A:POKE16527,B:FORA=CTOC+PL-1:READB:POKEA+(A>32767)*65
536,B:NEXT
9999 POKE16561,QY:POKE16562,QZ:CLEAR50:POKE16396,201:END
```

Program Listing

SAVE / add-ons for TRS-80® Software and Hardware

NEWDOS 80

A new enhanced NEWDOS for the TRS-80.

The most powerful Disk Operating System for the TRS-80, designed for the sophisticated user and professional programmer who demands the ultimate.

NEWDOS/80 is the planned upgrade from NEWDOS 2.1. Some of the features are:

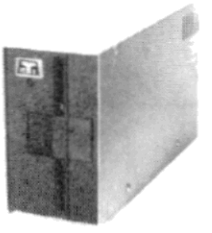
- New BASIC commands for files with variable record lengths up to 4095.
- Mix or match drives. Use 35, 40 or 80 track 5" disk drives or 8" disk drives, or combo.
- Security boot-up for BASIC or machine code application programs.
- New editing commands.
- Enhanced RENUMber that allows relocation.
- Command chaining.
- Device handling for routing to display and printer simultaneously.
- DFG function; striking of D, F and G keys allows user to enter a mini-DOS without disturbing program.
- Compatible with NEWDOS & TRSDOS.
- Machine language Superzap/80 2.1 utilities and enhanced debug and copy. **$149**

NEW

TF-8 80 TRACK DISK DRIVE **$639**
Double Your Capacity

TF-9 DUAL 80 TRACK DISK DRIVE **$799**
Quadruple Your Capacity

FACTORY CLEARANCE



Demo single or dual head MPI disc drive, complete with Power Supply and Chassis. Full warranty.

TF5D 40 TK **$279**

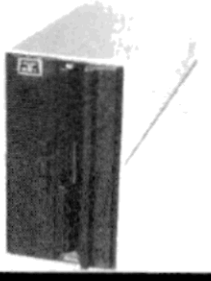
TDH-1D Dual 35 TK **$419**

LIMITED QUANTITIES

Announcing

8" Floppy Disk Drive System

for Model I



- One SA801 Floppy
- NEWDOS 80
- Cable & Adapter

$1095

Disk Drive Sale!

Save add-on for TRS-80

TF-3 Shugart SA400	$349
Pertec FD200, 40 track	$389
TF 5 MPI B51, 40 track	$379
TF7 Micropolis 77 track	$595
TDH-1 Dual Sided drive 35 track	$499
TF-3M Drive Sys. 2 Shugart	$698
NEWDOS+ 40 track $110 35 track	$99
Microconductor, Data Base Mgr. Mod I $249 Mod II	$399
AJA Business Pgk.	$289
Disk Head Cleaner	$16.95
Disk Alignment Pgm.	$109

Disk Expansion System

- 2 Shugart SA400 TF-3 ..... $718
- 1 Two-Drive Cable ..... $ 25
- 1 Expansion Interface 32K $489
- 1 35-track DOS+ ..... $ 99

TOTAL LIST PRICE $1331

SPECIAL PRICE ONLY $1,179

Same as above but includes TRS-80® Level II **$1984**

MOD II 8" Disk System

- 1 Drive System **$899**
- 2 Drive Expansion System **$1,399**

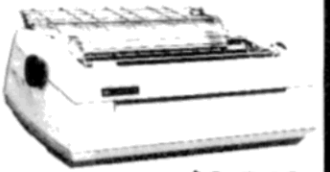
Drives for any Microcomputer

Does not include power supply & cabinet.

Pertec FD200	$282	FD250	$399
Shugart SA400	$279	SA800/801	$479
MPI B51	$279	B52	$349

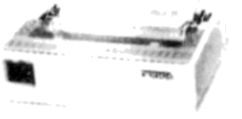
Printers

Centronic 779	$1,069
Base 2	$599
Centronic 737	$899
Centronics 702-9	$1,895
Anadex 9501	$1,595
Malibu	$2,395



Spinwriter **$2,549**

More Savings



INTRODUCTORY OFFER
SAVE $300
TRS-80 Graphics
List $1099

OKIDATA MICROLINE 80 **$649**

Memory Kit (16K)	$49.00
AC Isolator (6 socket)	$47.95
Modem	$164.00
Diskettes (10) Verbatim	$30.00

Apparat, Inc. ✓264

4401 South Tamarac Parkway • Denver, CO 80237 • (303) 741-1778

MICROCOMPUTER TECHNOLOGY INCORPORATED ✓28

3304 W. MacArthur • Santa Ana, CA 92704 • (714) 979-9923

TELEX 678-401 TAB IRIN

All prices cash discounted / Freight: FOB factory. Ask for our free catalog.

A handshaking story with a hard copy conclusion.

H-14, Meet the TRS-80

Frank Friesen
23 Gemini Avenue
Winnipeg, Manitoba
Canada R2G 0T5

After owning my TRS-80 for a while I became tired of copying programs from the screen by hand. I needed a printer. The H-14 line printer from Heathkit seemed to be the best buy. The kit cost less than half the amount of the Radio Shack line printer and even had lowercase.

The kit took only 12 evenings to finish, but while building it I realized that the on board CPU and software controlled UART could not easily be bypassed, as I had hoped. This meant that the only way I could communicate with the printer was by serial means.

The RS 232 or 20ma current loop could be bypassed so that information could come, via TTL levels. The problem was the expansion interface which puts out parallel information. This wasn't going to stop me—the price difference in printers could pay for a mini disk drive.

Looking around my workshop I saw an MM5303 general purpose UART (Universal Asyncronous Receiver Transmitter). Using this IC, a 74LS00 quad 2 input NAND gate and some extra wiring to the Heathkit control board, I made a working interface.

Circuit Operation

The circuit is simple and easy to understand (see Fig. 1). Parallel information is input to the UART (IC 1) from the expansion interface line-printer-port edge connector. Upon receiving the strobe on pin 23, the UART begins transmitting this data serially on pin 25 (SIN).

The format will be eight bits, with no parity, and one stop bit, as set by the voltage levels on pins 35 to 38 of IC 1. During this transmission, the COMPLETE signal, pin 24, of the UART, goes low. It causes pin 11 of IC 2 to go high, indicating to the computer that the printer, or at least the interface circuit, is busy.

When transmission is complete, pin 24 of the UART goes high, causing pin 11 of IC 2 to go low, indicating the printer is ready to receive more information.

The circuit derives its power from the H-14 line printer. Also the CLK signal on pin 40, which is used by the UART for timing and clocking the data, is taken directly from the UART in the H-14. This means that any baud rate set in the H-14 will be automatically set in the interface as well (not including 110 baud, since this requires two stop bits, not one as is set now). I run mine at 4800 baud with no problems.

The H-14 provides one handshaking signal, RTS, which goes low if the printer is off line or the print buffer is full. This signal is gated through IC2 to the line printer port. In this way, either the printer or the UART can cause a PRINTER BUSY signal.

The strobe signal is run through two gates simply to give the data lines a little additional time to stabilize before being strobed. Pin 23 on the edge connector is tied to pin 24 (GND) to disable the fault detect line since it is not needed.

Construction

I built the circuit on a pre-etched experimenter board that holds both IC's. I also used sockets to avoid unsoldering problems in the future.

Set a common +5 volt area and a common GND area and make the following connections:

1. Pins 3, 21, 36, & 39 of IC 1 to the common GND
2. Pins 1, 38, 37, 35 & 34 of IC 1 to the common +5 volts
3. Pin 24 of IC 1 to Pin 13 of IC 2
4. Pin 23 of IC 1 to Pin 6 of IC 2
5. Pins 3, 4 & 5 of IC 2 TOGETHER
6. Pin 14 of IC 2 to +5 volts
7. Pin 7 of IC 2 to GND

All that remains now is to wire the two interconnection cables. I made the cable to the H-14 out of six three-foot pieces of #24 stranded wire simply because I had them available. Six-conductor ribbon cable would also be suitable. Using different colored wires can make identification easier.

Before starting to wire the H-14, remove the jumper wire from J114,J115 or J114,J113. This interface circuitry is bypassed and the associated ICs could be removed (U101, U102, U103, U104).

Remove the bottom plate of the H-14 and make the following connections to the control board, using the six-wire cable. Be sure to count the pins correctly. Viewing from the bottom puts pin 1 on the opposite side. Also, be careful not to make any accidental solder bridges to other pins.

The U numbers refer to H-14 ICs and the others to the ICs on the board just completed.

1. One wire to IC U105 Pin 40;

the other end to the +5 volt common.

2. One wire to IC U105 Pin 32 (RTS); other end to IC 2 Pin 12.

3. One wire to IC U105 Pin 20; other end to common GND.

4. One wire to IC U105 Pin 15 (CLK); other end to IC 1 Pin 40.

5. One wire to IC U105 Pin 10 (SIN); other end to IC 1 Pin 25.

6. Last wire to IC U104 Pin 1 (−12); other end to IC 1 Pin 2.

This completes the wiring to the H-14.

Run the cable through the slot in the base plate. Re-install the base plate. Be sure the cable does not interfere with the print head movement.

For the second cable I used a ten-conductor ribbon cable about 15 inches long. I could not find a 34-pin female connector to mate with the expansion interface card edge, so I used a 40-pin connector and simply cut off the pins I did not need.

The pin configuration of the line printer port is shown in the expansion interface manual. Pin 1 is the pin on the top nearest the keyboard. Pin 2 is directly opposite on the bottom side of the board. Wire the cable as follows:

1. Connect one wire to Pin 1 of the connector; the other end to Pin 1 and 2 of IC 2.

2. One wire to Pin 2 of the connector; the other end to the common GND of the circuit board.

3. One wire to Pin 3 of the connector; the other end to Pin 26 of IC 1 (MM5303).

4. One wire to Pin 5 of the connector; the other end to Pin 27 of IC 1.

5. One wire to Pin 7 of the connector; the other end to Pin 28 of IC 1.

6. One wire to Pin 9 of the connector; the other end to Pin 29 of IC 1.

7. One wire to Pin 11 of the connector; the other end to Pin 30 of IC 1.

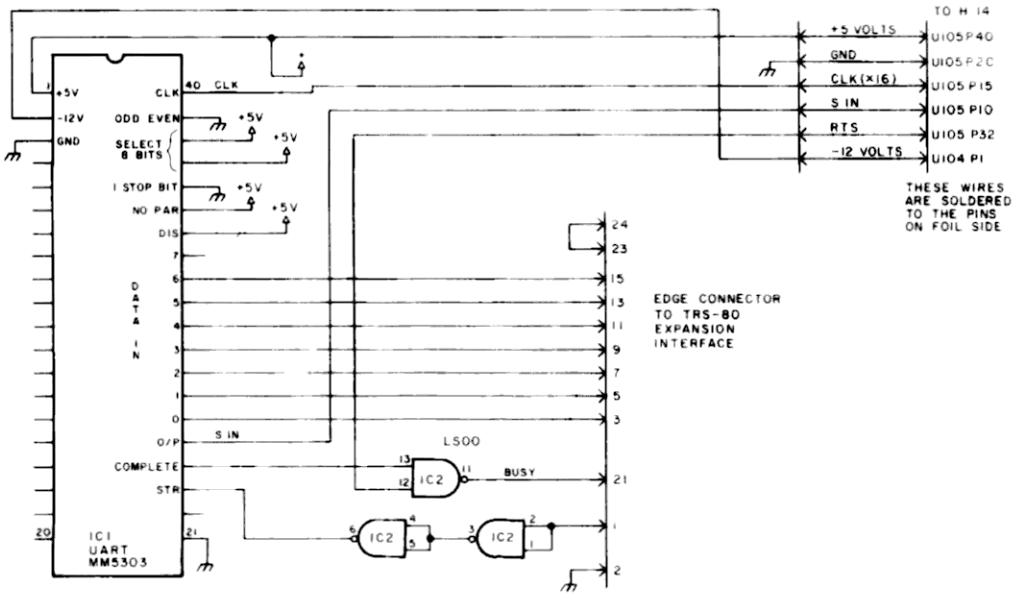


Figure 1.

8. One wire to Pin 13 of the connector; the other end to Pin 31 of IC1.

9. One wire to Pin 15 of the connector; the other end to Pin 32 of IC 1.

10. One wire to Pin 21 of the connector; the other end to Pin 11 of the IC 2.

11. Connect a small bare wire between Pin 23 and 24 on the connector itself.

Conclusion

I put the completed interface into a small experimenter box and ran one cable out each side. As seen, the parts count is very low. Parts required are:

One MM5303 (or equiv) UART

One 74LS00 Quad 2 I/P NAND GATE

One 40-pin socket (recommended)

One 14-pin socket (recommended)

One 34-pin female connector (0.1-inch spacing)

Miscellaneous wire, PC board, etc.

The cost of the unit is less than $15.

I normally operate the H-14 using the manual switch to select character width. Because the printer driver routine only sends carriage returns (not line feeds), the dip switch (sw102) in the H-14, position 3, should be set to 0, causing an automatic line feed on carriage return. This works fine until I try to select a new character width using this command:

```
LPRINT CHR$(27); CHR$(117); CHR$(20)
```

The above command causes 96 characters per line. The only problem is that the H-14 will no longer generate internal line feeds. The easiest way around this is to generate a line feed from BASIC in a subroutine such as the following:

```
XXX PROGRAM LPRINT LINE: GOSUB
10000
10000 IF PEEK(14312)<>63 THEN 10000
ELSE POKE14312,10: RETURN
```

Line 10000 waits until the printer is ready and then sends a line feed directly, bypassing the print driver routine.

If you change character width by software, calling this line in a program after the LPRINT line produces the line feed needed.

It is easy to see that this will not work with LLIST or LPRINT lines longer than the number of characters available on the line. To use LLIST you must use the manual mode and are limited to only the small or large characters. Note that the dip switch is sampled only at power up. This means if you change character size with software, you must turn the H-14 off and on again to get back the auto line feed function. If you use large characters, this is only a minor inconvenience.■

THE PROGRAM STORE (202) 337-4691

4200 Wisconsin Ave NW PO Box 9609 Washington D.C. 20016

All programs for TRS-80 16k, Level II computer.

NEW!

microcomputer games®
from The Avalon Hill Game Company

Avalon Hill Game Company has just introduced their first five war and strategy games for the home computer. You play against the program. Each package includes instructions and software for the TRS-80, APPLE II and PET computers having 16k of memory.

Midway Compaign
The battle of Midway is recreated with you in control of the outnumbered and outranged U.S. Navy. The Japanese need air superiority to win. $14.95

B-1 Nuclear Bomber
Pilot your advanced bomber towards the target city in the Soviet Union. Avoid the MiG fighters and the surface-to-air missles. $14.95

North Atlantic Convoy Raider
This simulates the Bismark convoy raid of 1941. You control the British Home Fleet. $14.95

Nuke War
This nuclear confrontation pits you against the computer. You choose either massive espionage or military build-up. $14.95

Planet Miners
One to four players compete against the computer staking claims in the solar system. Watch out for sabotage and claim jumping. $14.95

INTERACTIVE FICTION

by Robert Lafore from Adventure Inter.

In these TRS-80 disk programs you influence the story by speaking with the characters. Each program sets a fictional scene. Then you start the dialogue with the other characters.

Six Micro Stories offers a good introduction. $14.95

Local Call For Death is a detective story in the style of Lord Peter Whimsey. $19.95

Two Heads of the Coin is a psychological mystery set in the London of Sherlock Holmes. The most challenging. $19.95

Electronic Handicapper BASKETBALL

by Sothen, Laurence & Gavenda
from Acorn Software Products

Basketball is the first in the Electronic Handicapper series from Acorn. It will introduce you to the benefits of predicting the winners of this season's basketball games. This two-tape package gives you power ratings to get you started. You keep the data tape informed of the current week's wins, losses and points. The program then calculates a winner and point spread for you to use. Last season it was used to predict 85% of the winners with a 64% accuracy with the point spread. 16k required. $99.00 on tape.

STOCK TRADER

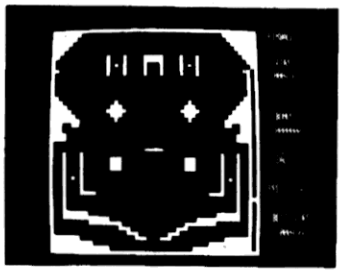
from Galactic Software

This system is designed for the active "trader" and not the long term investor because the system is technically oriented. It tracks issues you select and reflects their performance against the overall market. There is also a comparison of the issue against itself to allow spotting "unusual" activity.

The initial data are from either the Standard and Poor Stock Guide or Value Line. The daily data of high, low, close and volume are input from the newspaper.

The program is intended to be a guide to indications and not as a sole recommendation.

Tape $89.00 Disk $99.00
Manual only $20.00



PINBALL

by John Allen from Acorn

Get your flipper fingers ready for action in this real-time, machine language game. Lots of sound and flashing graphics. There are five speeds so anyone can play. This version features the dreaded "Bermuda Square"!
Protected tape $14.95
Protected disk $20.95

INVADERS FROM SPACE

by Carl Miller from Acorn

"Maybe it's too fast!" Perferred by all the local arcade addicts, this machine language game has great sound. Alien Invasion, Invaders and Invaders Plus just can't compete. You can adjust parameters including the speed, if you think it's too fast. Only a few heroes will be able to save earth.
Protected tape $14.95
Protected disk $14.95

THE EMPIRE STRIKES!

from Computer Simulations Company

The rebellion begins with one base and one warship. You take on fighters, conduct ground operations and secure planets, adding to the number of Rebel bases. Don't let any Empire Scoutcraft escape! You are the last hope. $14.95

SUPERSCRIPT

by Richard Wilkes from Acorn

"Scripsit" from Radio Shack is a great program, but it lacks some features. SuperScript adds features to your disk version of "Scripsit." Some of these features are:

You can get a directory or kill files from within SuperScript. TRSDOS or NEWDOS can read SuperScript files. You can insert text into unjustified lines during printout. For example, inserting a name after "Dear" and before the colon. For this purpose a lowercase driver is included.

On printers that can backspace, underlining and slashed zeroes (ø) are options. On Diablo and NEC printers one can superscript, subscript, underline, boldface and select 10/12 pitch.

The keyboard driver is changed to allow a correct key repeat which is faster than tapping on a key and which does not destroy the video display. The initial character sent to the prnter is changed from a linefeed to a carriage return to empty the buffer. A required space may be specified when it is undesirable to place spaces between parts of text when justifying. From the keyboard you can also enter special characters such as brackets, braces and carets.

Serial and parallel drivers are included on the disk. You can customise these drivers for use with other types of letter quality printers. The serial drivers are included which use the ETX/ACK protocol for 1200 baud communications. Furthermore, printer drivers can be protected in high memory.

The "L" command used to load a file now requires a filespec to avoid destroying text buffer if the question mark is omitted from the "?L" command.

On disk for $29.95

STUCTURED BASIC TRANSLATOR

by Gene Bellinger from Acorn

Try structured programming. You can write programs using PROCEDURES, CALLS, CASE--CALLS, IF-THEN-ELSE, WHILE and UNTIL. Once written, SBT will quickly translate the structured code into an efficient BASIC program. Speeds up program development and documentation. The program is both fast (a 20k BASIC program in less than 4 minutes) and compact. Requires 32k and one disk drive. Supplied on disk for $29.95

THE PROGRAM STORE
4200 Wisconsin Ave NW
P.O. Box 9609 Dept. K 4
Washington DC 20016

VISA AND MASTERCARD

TO ORDER TOLL FREE 800-424-2738

For program information call (202) 337-4691

Mail orders: Send check or money order for total purchase price, plus $1.00 for shipping & handling. Charge card customers, please include card number, expiration date, and your name as it appears on the card.

A sad tale of one user's efforts at homebrew interfacing.

Caveat Emptor

M. Parris
646 Island Park Drive
Ottawa, Ontario
Canada K1Y 0B7

When I bought a TRS-80 it was with the expectation of using it with a hard copy device of some kind, and, until line-printer prices became more reasonable, this was almost certainly going to be the beat up Olivetti terminal I use—110 baud, RS232 compatible.

However, I didn't want to eliminate the option of eventual higher printing rates, so I looked around for some type of general purpose I/O interface. Radio Shack's RS232 board together with the necessary expansion interface cost a minimum of $400. On the other hand, with some pain, I might be able to design and build an I/O interface for about $35.

Between these two extremes, the best bet appeared to be the TRS-80 Serial I/O board kit offered by Electronic Systems for $59.95 (connecting cable $19.95 extra), advertised with variable baud rate, variable bit count and parity and LPRINT, LLIST and BASIC input. My estimate of $35 was based on Electronic Systems' parts list and current catalog prices, but the thought of hours of designing and fabricating a board was enough to persuade me to try the kit.

Still Waiting

I sent away for the kit and waited over seven weeks for its arrival. Unpacking it, I was concerned that Electronic Systems had not taken any antistatic precautions. The components, jumbled together in a polyethylene bag, might have zapped some of the low power logic, diode-protected or not. Several of the I.C. and socket pins were bent as a result of this packaging, and there was no component list (other than that appearing in the catalog).

Documentation consisted of a circuit diagram with no component values marked, a component placement diagram and two short driver routine listings, together with a note concerning baud rate, parity, data and stop bits selection via onboard dip-switches.

Ordinarily, the lack of component values wouldn't matter, if an accurate component placement diagram were included. However, in this case there was a discrepancy between the parts supplied and the parts necessary to assemble the board as per diagram. This particular hurdle was crossed by means of a few educated guesses and the purchase of one extra resistor. I *did* write to Electronics Systems by the way, enclosing an S.A.S.E. too, but didn't get any clarification.

There was need for some caution in attaching the 40-conductor cable and its socket. The pin numbers were marked on the board in such a way that they appeared to reference the expansion port connector, in which case I might have easily connected the cable upside down. It's advisable to make certain (and mark the orientation) of cable connectors, board and TRS-80 expansion port before switching on the soldering iron!

The remainder of the assembly was easy. I'd decided that I might eventually want to introduce some subtleties into the UART's handshake, so I reinstated the switching arrangement which Electronic Systems had apparently removed in this revised version of the I/O board. This involved mounting a dip-switch (the board was already drilled for it anyway) and cutting the foil at the five places where the switch elements had been strapped closed. Functionally, the circuit is unaltered provided the switches are closed.

No Reply

Assembly complete, two clock adjustments had to be made for the baud rates. The 110 baud adjustment (1760 hz) was an easy matter, but the 150-2400 baud clock (38.4 khz) proved intractable and the trouble was eventually traced to a faulty gate in a CD4096—possibly static damaged.

Once again I wrote to Electronic Systems, suggesting they might like to send a replace-

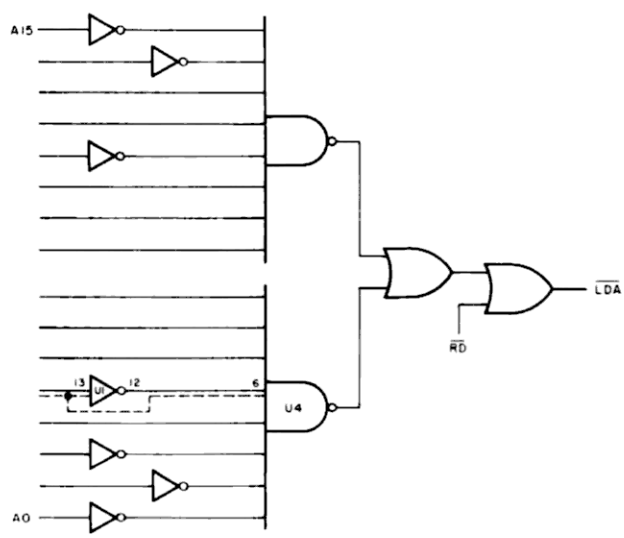


Figure 1.

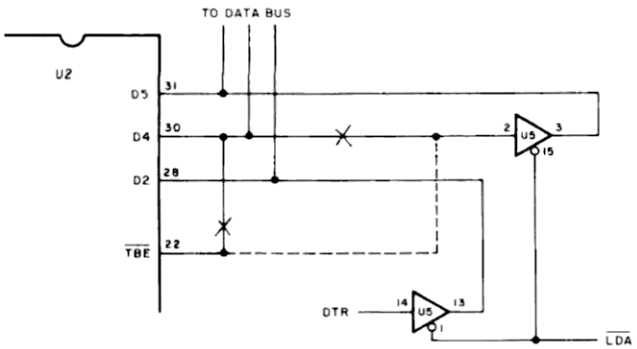


Figure 2.

ment—to date, though, I've had no reply!

In any event, I replaced the CD4096, successfully adjusted the clock and burned in the board for a few hours. The board required about 130 mA, +5V. and 50 mA, −12 V. This would have been nice to know beforehand, but it was about what I'd expected and I'd already made the power supply anyway!

By this time I was more than a little concerned at the poor documentation supplied by Electronic Systems and thought it advisable to read over the routines supplied, one for LLIST, LPRINT, the other for BASIC input. Disassembly of the programs showed that the board's address was treated as 37F8 hex. Fig. 1 shows the address decoding logic in the circuit diagram which was supplied, and the address is clearly 37E8 hex, as, after all, it should be for the line printer driver routine in ROM.

But why capriciously use custom written I/O routines when there's an adequate one already in ROM?

I reckoned I should check the whole board against the circuit diagram. Among the several discrepancies revealed, which would have made troubleshooting from the diagram difficult or impossible, the most important was that A4 was not inverted. U1 was simply bypassed at pins 12 and 13 as indicated in Fig. 1. The board address was, after all, 37F8 hex and not 37E8 hex, making it impossible to use the board without Electronic Systems' routine.

The hardware fixup is easy enough. Cutting out the foil bypass and reconnecting U1 pin 12 to U4 pin 6, the board address is restored to 37E8 hex. As a temporary, but less satisfactory expedient, U4 pin 6 can be isolated by bending the I.C. pin out so that it doesn't insert into the socket, which sets A4 permanently high in the board address. A better method is to use a two pole, two way nonshorting switch to preserve compatibility with Electronic Systems' I/O routines.

Diagram Discrepancy

Fig. 2 shows the other major circuit diagram discrepancy; neither U2 pin 22 ($\overline{TBE}$) or U5 pin 2 is connected to U2 pin 30, but they are connected together. The UART status is detected via U2 pins 31 and 28 only (D3 and D5), so that the board can't be used with the TRS-80 in-ROM line printer driver routine without some recoding of its status.

In Fig. 3 I've shown the changes I made, in addition to the board address change of course, in order to use in-ROM LLIST, LLPRINT without custom written software.

```
                00100 ;AUTO LF FIXUP FOR RADIO SHACK EDTASM
                00110 ;USING ELECTRONIC SYSTEMS SERIAL I/O
                00120 ;LOAD AFTER EDTASM
                00130 ;/18058 TO RUN
37F8            00140          ORG 37F8H        ;BOARD ADDRESS
37F8            00150 INOUT    EQU $            ;SET ITS LABEL
0024            00160 STMASK   EQU 24H          ;UART USES D3,D5
0024            00170 STDATA   EQU 24H
45CB            00180          ORG 45CBH        ;CHANGE ADDRESS HERE
45CB F837       00190          DEFW INOUT
45D9            00200          ORG 45D9H        ;CHANGE ADDRESS HERE
45D9 F837       00210          DEFW INOUT
45DE            00220          ORG 45DEH        ;CR COMING UP
45DE C3D448     00230 GOFIX    JP FIXUP         ;SO FIRST PUT OUT LF
45E1            00240 BACK     EQU $            ;RETURNING TO HERE
45EF            00250          ORG 45EFH        ;CHANGE ADDRESS HERE
45EF F837       00260          DEFW INOUT
45F1            00270          ORG 45F1H        ;CHANGE UART STATUS
45F1 E624       00280          AND STMASK       ;CHANGE THE MASK
45F3 FE24       00290          CP STDATA        ;CHANGE THE DATA
48C9            00300          ORG 48C9H        ;PUT THE PATCH HERE
48C9 45         00310          DEFB 'E'         ;"EDTASM HERE"
48CA 44         00320          DEFB 'D'
48CB 54         00330          DEFB 'T'
48CC 41         00340          DEFB 'A'
48CD 53         00350          DEFB 'S'
48CE 4D         00360          DEFB 'M'
48CF 20         00370          DEFB ' '
48D0 48         00380          DEFB 'H'
48D1 45         00390          DEFB 'E'
48D2 52         00400          DEFB 'R'
48D3 C5         00410          DEFB 'E'+80H     ;EOM
48D4 CDEE45     00420 FIXUP    CALL 45EEH       ;CHECK THE UART
48D7 20FB       00430          JR NZ,FIXUP      ;IF BUSY, WAIT
48D9 3E0A       00440          LD A,0AH         ;OTHERWISE, LF
48DB 32F837     00450          LD (INOUT),A     ;PUT OUT CHARACTER
48DE DD3404     00460          INC (IX+4)       ;REPLACE THE OPCODE
48E1 C3E145     00470          JP BACK          ;CONTINUE AS BEFORE
0000            00480          END
00000 TOTAL ERRORS
```

Listing 1.

```
                00100 ;LINE PRINTER DRIVER FOR TRS-80
                00110 ;USING ELECTRONIC SYSTEMS SERIAL I/O
                00120 ;*/6681* AND THEN *CLEAR* FOR BASIC
37F8            00130           ORG 37F8H       ;BOARD ADDRESS
37F8            00140 INOUT     EQU $           ;SET ITS LABEL
0024            00150 STATM     EQU 24H         ;SET STATUS MASK
0024            00160 STATD     EQU 24H         ;SET STATUS DATA
7F9E            00170           ORG 7F9EH       ;PUT THE ROUTINE HERE
7F9E            00180 LINEP     EQU $           ;LABEL FOR DCB
7F9E 79         00190           LD A,C          ;GET CHARACTER
7F9F B7         00200           OR A            ;NULL?
7FA0 2856       00210           JR Z,STATUS     ;YES, CHECK UART
7FA2 FE0B       00220           CP 0BH          ;VERT. TAB?
7FA4 280A       00230           JR Z,VTAB       ;YES, DO IT
7FA6 FE0C       00240           CP 0CH          ;NEW PAGE?
7FA8 201C       00250           JR NZ,PRINT     ;NO, PRINT CHARACTER
7FAA AF         00260           XOR A
7FAB DDB603     00270           OR (IX+3)       ;GET PAGE SIZE
7FAE 2816       00280           JR Z,PRINT      ;OOPS!
7FB0 DD7E03     00290 VTAB      LD A,(IX+3)     ;PAGE SIZE AGAIN
7FB3 DD9604     00300           SUB (IX+4)      ;GET LINES LEFT
7FB6 47         00310           LD B,A          ;INTO B
7FB7 CDF87F     00320 WAIT1     CALL STATUS     ;CHECK UART
7FBA 20FB       00330           JR NZ,WAIT1     ;IF BUSY, WAIT
7FBC 3E0A       00340           LD A,0AH        ;OTHERWISE, LF
7FBE 32F837     00350           LD (INOUT),A    ;PUT IT OUT
7FC1 10F4       00360           DJNZ WAIT1      ;UNTIL PAGE DONE
7FC3 4F         00370           LD C,A          ;STORE CHARACTER
7FC4 1822       00380           JR RESET        ;THEN RESET COUNT
7FC6 F5         00390 PRINT     PUSH AF         ;SAVE CHARACTER
7FC7 CDF87F     00400 WAIT2     CALL STATUS     ;CHECK UART
7FCA 20FB       00410           JR NZ,WAIT2     ;BUSY?
7FCC F1         00420           POP AF          ;GET BACK CHARACTER
7FCD 32F837     00430           LD (INOUT),A    ;PUT IT OUT
7FD0 FE0D       00440           CP 0DH          ;CR?
7FD2 C0         00450           RET NZ          ;IF NOT, DONE
7FD3 CDF87F     00460 WAIT3     CALL STATUS     ;CHECK UART
7FD6 20FB       00470           JR NZ,WAIT3     ;BUSY?
7FD8 3E0A       00480           LD A,0AH        ;LF TOO
7FDA 32F837     00490           LD (INOUT),A    ;PUT IT OUT
7FDD DD3404     00500           INC (IX+4)      ;UPDATE COUNTER
7FE0 DD7E04     00510           LD A,(IX+4)     ;ROOM LEFT?
7FE3 DDBE03     00520           CP (IX+3)       ;SO CHECK SIZE
7FE6 79         00530           LD A,C          ;GET BACK CHARACTER
7FE7 C0         00540           RET NZ          ;RETURN IF ROOM
7FE8 CDF87F     00550 RESET     CALL STATUS     ;CHECK UART
7FEB 20FB       00560           JR NZ,RESET     ;BUSY?
7FED 3E0A       00570           LD A,0AH        ;DOUBLE SPACE
7FEF 32F837     00580           LD (INOUT),A    ;TO MARK PAGE
7FF2 79         00590           LD A,C          ;GET BACK CHARACTER
7FF3 DD360400   00600           LD (IX+4),0     ;RESET COUNTER
7FF7 C9         00610           RET             ;RETURN
7FF8 3AF837     00620 STATUS    LD A,(INOUT)    ;LOOK AT UART
7FFB E624       00630           AND STATM       ;STATUS MASK
7FFD FE24       00640           CP STATD        ;STATUS DATA
7FFF C9         00650           RET             ;ZERO SET IF OK
4026            00660           ORG 4026H       ;CHANGE DCB
4026 9E7F       00670           DEFW LINEP      ;AT DRIVER ADDRESS
0000            00680           END
00000 TOTAL ERRORS
```

Listing 2.

Hold it though! Even if these amendments are applied, there may yet be a problem. The in-ROM line printer driver routine assumes you have a line printer with auto line feed! If a line feed character (0A hex) is encountered, it's replaced by a transparent null character (00 hex). (See the TRS-80 BASIC II ROM 39C-3C1 hex (924-961 decimal.) Unless your hard copy device is equipped with auto line feed or your serial I/O board is designed to deal with the problem you're out of luck!

Many other purchasers of Electronic Systems' TRS-80 Serial I/O Interface must have encountered these problems. To

```
 1 REM LPRINT, LLIST FOR 16K TRS-80, MEM SIZE = 32670
 2 POKE16422,158:POKE16423,127
 3 FORI=32670TO32767:READA:POKEI,A:NEXT
 4 DATA121,183,40,86,254,11,40,10
 5 DATA254,12,32,28,175,221,182,3
 6 DATA40,22,221,126,3,221,150,4
 7 DATA71,205,248,127,32,251,62,10
 8 DATA50,248,55,16,244,79,24,34
 9 DATA245,205,248,127,32,251,241,50
10 DATA248,55,254,13,192,205,248,127
11 DATA32,251,62,10,50,248,55,221
12 DATA52,4,221,126,4,221,190,3
13 DATA121,192,205,248,127,32,251,62
14 DATA10,50,248,55,121,221,54,4
15 DATA0,201,58,248,55,230,36,254
16 DATA36,201
```

Listing 3.

recap: I had a serial I/O interface that worked splendidly provided that I could use custom-written driver routines or that it could be made compatible with the TRS-80 ROM or Radio Shack software; but only if I had an auto line feed printer, which I didn't.

Listing 1 shows minimum changes which must be made to Radio Shack's Editor/Assembler program in order to make it directly compatible with Electronic Systems' TRS-80 Serial I/O Interface. Simply load the object tape of this routine after the EDTASM tape and run the program at 18058 decimal.

Listing 2 shows a comprehensive line printer routine which completely replaces the in-ROM routines for LLIST, LPRINT more effectively than the (much shorter) routines Electronics Systems provided.

Listing 3 shows a BASIC loader program, the equivalent of Listing 2.

Conclusion

My final verdict is that this serial I/O board will—just barely—do what the manufacturers claim for it. Within these limits it works well, and can be made to work better. The design, however, was badly thought out and some customers might have trouble.■

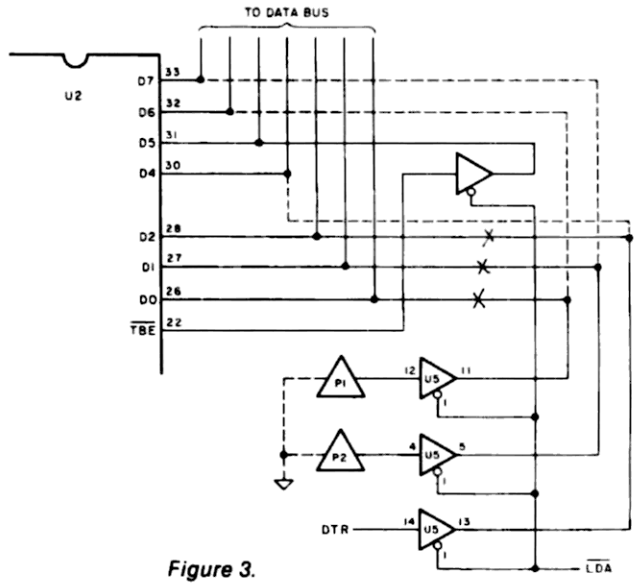


Figure 3.

H&E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

★ *All Orders processed within 24-Hours*
★ *30-Day Money Back Guarantee on all Software (less a $3 penalty for handling)*
★ *10-Day Money Back Guarantee on Disk Drives and Printers* **PLUS** *120-Days Free Service*

- **LEARNING LEVEL II** By David Lien
  The Original Author Of The Level Manual
  A Step By Step approach to Learning Level II especially geared to new TRS-80™ Owners
  $15.95

- **TRS-80™ DISK AND OTHER MYSTERIES**
  Over 100 pages of indespensible information for disk owners. Learn to recover information from bad disks, how to make Basic programs unlistable and 12 more chapters of never published tips and information. Written by H.C. Pennington.
  (For all Disk Owners) **$22.50**

NEW SBSG BUSINESS SYSTEM FOR MODEL I OR MODEL II - IN STOCK
- General Ledger
- Accounts Receivable
- Accounts Payable
- Payroll
- Inventory Control with Invoicing

- Each module can be operated individually or as a coordinated **SYSTEM.** Turn-Key error catching operation for beginners.
- Complete manual and documentation accompany each program.
- Minimum System requirements 2-Disk Drives for Model I...1-Disk Drive for Model II
- Each module can be formatted to span data on up to 4-Disk Drives.
- Free 30-Day telephone consultation
- Call for complete specifications
- Model I Version **$125.00** Per Module, **$495.00** Per System
- Model II Version **$225.00** Per Module, **$995.00** Per System

DATA MANAGEMENT SYSTEMS
- DMS replace index cards or any data requiring long lists of information
- TBS In-Memory Information System (For Cassette Systems) **$39.95**
- TBS Disk Data Manager (Requires 1 or more disk drives)...Set up fast random access files in minutes. Stores up to 320K of information on 4 Drives. Up to 10 fields and 255 characters per record. Supports upper and lower case. RS-232 or TRS-232...Features complete editing **$49.50**
- Personal Software CCA Data Management System...Completely user oriented, menu drive, 130 page Step By Step Manual...Capable of inventory control, sorting data, reporting data in nearly any form (for reports and mailing labels). Sorts data by up to 10 fields for zip code, balance due, geographic location or whatever. Prints reports with subtotals and totals automatically calculated. Fast random access **$75.00**

FROM RACET COMPUTES
- **REMODEL-PROLOAD** - Renumbers program lines, combines programs. The only renumber program that will renumber the middle of a program. Specify 16K, 32K or 48K. Works with Cassette or Disk **$34.95**
- **GSF** - Use in your Basic Programs for Instant Sorting (will sort 1000 items in 9 seconds). Other commands include Compress and Uncompress Data, Duplicate Memory, Display Screen Controls and Fast Graphic Controls **$24.95**
  (For Cassette or Disk, specify 16K, 32K or 48K)
- **DOSORT** - All G.S.F. commands plus special Multiple Disk Sorting Routines **$34.95**
  (Specify 32K or 48K)
- **INFINITE BASIC** - Adds 70 commands to your TRS-80™ including Instant Sort, Matrix Commands, String Commands, Left and Right Justification, String Centering, Simultaneous Equations, Upper and Lower Case Reverse and more. (For Cassette or Disk) **$49.95**
- **INFINITE BUSINESS** (Requires Infinite Basic) Eliminate Round-off error, 127-Digit Calculation Accuracy, Insert New Elements in Sorted Arrays, Automatic Page Headings, Footings and Pagination, Multiple Precision Arithmetic and more. (For Cassette or Disk) **$29.95**
- **COPSYS** - Copy Machine Language Programs (For Cassette Only) **$14.95**
- **DSM** (Disk Sort Merge) **$75.00**

FROM SMALL SYSTEM SOFTWARE
- **RSM-2** Machine Language Monitor **$26.95**
- **RSM-2D** Disk Version of RSM-2 **$29.95**
- **DCV-1** Converts Machine Language Programs from tape to disk **$9.95**
- **AIR RAID** - The ultimate TRS-80™ game converts your TRS-80™ into a real time shooting gallery **$14.95**
- **BARRICADE** - A fast pong style game **$14.95**
- **CPM** - (For Disk Only) **$150.00**
- **TRS-232 INTERFACE** - Interface with Software driver RS-232 printers to your TRS-80™ **$49.95**
- **TRS-232 FORMATTER** - Additional (optional) Software for TRS-232 owners. Adds many printer commands to your TRS-80™ **$14.95**
  (With purchase of TRS-232) **$9.95**
- **PENMOD** - Use the Electric Pencil with RS's lower case modification **$19.95**

FROM GALACTIC SOFTWARE
- **MAIL PAC** - For Model I Disk Systems only **$99.95**
  Quick-sorting full user control over mailing list from Galactic Software.
- **STOCK MARKET PAC** **$99.95**

FROM APPARAT NEW DOS **$99.95**
35, 40 and 77 Track Versions available
- **NEW DOS/80** (With variable record length files, chainings and many other features **$149.95**

FROM THE BOTTOM SHELF
- **CHECKBOOK II** (For Cassette or Disk) **$39.95**
- **SYSTEM DOCTOR** (A complete diagnosis of your TRS-80™. Checks memory, video, cassette, disk, ROM, and all other parts of your system)
  For Cassette or Disk **$28.50**
- **CHECKBOOK REGISTER ACCOUNTING SYSTEM** (Requires 2 disk drives) **$75.00**
- **LIBRARY 100** - 100 established business, game and educational programs plus **FREE** Tiny Pilot all for **$49.50**
- **BASIC TOOL KIT** - Lists all variables, GOTO's and GOSUB's in your program **$19.80**

- **SOUNDWARE** - Adds sound to your TRS-80™ Just plug it in **$29.95**
  Sample programs included
- **TING TONG** - Can be used with Soundware for a **Sound** version of pong **$9.95**

- **VIC - The Carta Visual Instructional Computer Program** **$19.95**
  The Level II 16K Cassette is designed to teach beginners the Basics of Machine Language and Assembly Language Programming. See every Machine Language Instruction Display on your video. VIC includes Step By Step 55 page manual.

VISTA V80 DISK DRIVE - 110K of Storage **$395.00**
Add $29.95 for Cable (Free with purchase of 2-Disk Drives) 10 day money back guarantee

FROM HOWE SOFTWARE
- **MON-3** - Machine Language Programming for beginners. MON-3 is a complete System Monitor with Users Manual **$39.95**
- **MON-4** - Disk Version of MON-3 **$49.95**

FROM MICROSOFT
- **LEVEL III BASIC** **$49.95**
  Now Cassette owners can add Disk Commands to their TRS-80™ without owning a Disk Drive
- **MICROSOFT DISK ADVENTURE** **$29.95**
- **TRSDOS BASIC COMPILER** **$195.00**
  Run Basic Programs up to 15 times faster

- **NEC BUSINESS QUALITY PRINTERS** (For MOD-I or MOD-II) **$2,995.00**

THE ELECTRIC PENCIL
Cassette **$99.95**
Disk **$150.00**
MOD-II Version **$325.00**

- **HORSE SELECTOR II** By Dr. Hal Davis
  The TRS-80™ version updated for the TRS-80™ and originally reviewed in Systems and Methods **$50.00**

•••EVERYTHING•••
FOR MOD-II OWNERS

NEW MOD-II NEWSLETTER	
MOD-II Catalog **Free** w/subscription	**$12/year**
MAIL PAC	**$199.95**
MICROSOFT BASIC COMPILER	**$395.00**
MICROSOFT BASIC	**$325.00**
GSF SORT ROUTINE	**$50.00**
CP/M	**$170.00**
PEACHTREE BUSINESS SOFTWARE	**Call**
WORD STAR	**$495.00**

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™ ↙9

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

HOURS: 9-5
Monday thru Saturday

24 HOUR ORDER LINE
(914) 425-1535

48-Page Catalog $2 **FREE** *With Any Order*
Order By Phone Or Mail
Add $1 Per Order For Shipping Within UPS Areas
Add $3 For C.O.D.
Add $3 For All Foreign And Non-UPS Shipments
Add $3 For UPS Blue Label

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

↙Reader Service—see page 226

(Continued on next page)

(PAYROLL CAPABILITIES CONTINUED)

- ★ employees can be paid using any combination of pay types (except, hourly cannot receive salary & salary cannot receive hourly)
- ★ special non-taxable or taxable lump sums can be paid regularly or one time (bonus, reimbursements, etc)
- ★ health & welfare deductions can be automatically calculated for each employee
- ★ earnings-to-date are accumulated and added to permanent records; taxes are computed and deducted US income tax, Social Security tax, state income tax, other deductions (regular or one time)
- ★ paychecks are printed; computer checks with your company letterhead can be purchased from SBSG
- ★ calculations are accumulated for; employee pay history, 941A report, W-2 report, insurance report, absentee report
- ★ fully linked to General Ledger. Each employee's payroll information can be distributed to as many as (12) twelve different GL accounts; system automatically posts to cash account

INVENTORY/CONTROL INVOICING

- OVER 1000 ITEMS ON MODEL I
- OVER 3000 ITEMS ON MODEL II
- LOW STOCK ALARM
- INVOICING DEDUCTS FROM INVENTORY
- COMPLETE INVENTORY REPORTS
- REORDER POINT REPORT
- QUICK ITEM ACCESS

CLIENT BILLING, STOCK CONTROL, DENTAL BILLING, COMMODITIES
Medicare/Medicaid billing also available

MODEL I **$125 Per Module**
$495 Complete System

MODEL II **$225 Per Module**
$995 Complete System

WE ARE THE ONLY SOFTWARE COMPANY THAT OFFERS A REFUND WITHIN 30 DAYS ON ALL SOFTWARE (H & E COMPUTRONICS MONTHLY NEWSMAGAZINE SUBSCRIBERS ONLY). WE DO CHARGE A $3 PENALTY TO COVER POSTAGE AND HANDLING.

GENERAL LEDGER

The General Ledger accounting system consolidates financial data from other accounting subsystems (A/R, A/P, Payroll, direct posting) in an accurate and timely manner. Major reports include the Income Statement and Balance Sheet and a "special" report designed by management. The beauty of this General Ledger system is that it is completely user formatted. You "customize" the account numbers, descriptions, and report formats to suit your particular business requirements. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now a well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES

- ★ more than 200 chart of accounts can be handled
- ★ account number structure is user defined and controlled
- ★ more than 1,750 transactions may be entered via
  - direct posting; done by hand, validated against the account file before acceptance
  - external posting; generated by A/R, A/P, Payroll or any other user source
- ★ data is maintained and reported by
  - month
  - quarter
  - year
  - previous three quarters
- ★ reports (samples on back) include
  - trial balances
  - income statement
  - balance sheet
  - special accounts reports and more
- ★ user formats reports with the following designed as you wish
  - titles
  - headings
  - account numbers
  - descriptions
  - subtotals
  - totals
  - skip lines
  - skip pages
- ★ up to eight levels of totals - fully user designated
- ★ menu driven, easy to use, full screen prompting and cursor control

COMPUTRONICS INC
MATHEMATICAL APPLICATIONS GROUP

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

PLEASE SEND ME:

MODEL I	$125 PER MODULE	____
	$495 COMPLETE SYSTEM	____
MODEL II	$225 PER MODULE	____
	$995 COMPLETE SYSTEM	____

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

CREDIT CARD NUMBER _____ EXP. DATE _____

SIGNATURE _____

NAME _____

ADDRESS _____ CITY _____ STATE _____ ZIP _____

*** ADD $6/YEAR (CANADA, MEXICO) - ADD $12/YEAR AIR MAIL - OUTSIDE OF U.S.A., CANADA & MEXICO ***

Reader Service—see page 226

H & E COMPUTRONICS INC.

...EVERYTHING FOR YOUR TRS-80™...

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

MICROSOFT BASIC COMPILER

With TRS-80™ BASIC Compiler, your Level II programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC interpreter are included in the package. Features like the WHILE...WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution. You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80™ Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation **$195.00**

1980 INCOME TAX PAC

Completely Revised · Latest Tax Tables · Fully Tested · Complete Manual and Documentation. The new version of the Income Tax Pacs are full of error catching codes making it impossible to make an error. Follow the simple Step By Step procedure that makes tax preparation simple.

INCOME TAX PAC A ..... **($19.95...Cassette)**
For Level II 16K Cassette Only
Does Form 1040 and 1040A
- Schedule A itemized deductions
- Schedule B interest and dividends
- Output to video display
- Schedule TC tax computation

INCOME TAX PAC B ..... **$49.95...Cassette or Diskette)**
For Level II 16K with or without printer...cassette or disk has all features of Income Tax Pac A **Plus** works with or without line printer.
- Formats Form 1040 and 1040A for standard tax forms
- Schedule C income from a personally owned business
- Form 2106 employee business expense

PROFESSIONAL INCOME TAX PAC C ..... **$99.95...Diskette**
For Level II 32K with disk and printer (optional)
Has all features of Income Tax Pac B **Plus** automatic memory storage for income tax preparers.
- 22 additional schedules and forms
- Formats forms for individual or tractor feed printing

MOD II CPA VERSION **$199.95**

GUARANTEED PROFIT **91%** **WINS PLACES SHOWS** **32%** **AVERAGE PROFIT AT ALL TRACKS-1978**

THE HORSE SELECTOR II (FLATS) (By Dr. Hal Davis **$50.00**
New simplified version of the original Horse Selector. The first Horse Selection System to actually calculate the estimated odds of each horse.
HIGHER PROFITS (OVER 100%) POSSIBLE THROUGH SELECTIVE BETTING ON:
- Rates each horse in 10 seconds.
- Easy to follow rules.
- Can be used with any Apple II Computer.
- 100% money back guarantee (returned for any reason).
- Uses 4 factors (speed rating, track variant, distance of the present race, distance of the last race).
- Using the above factors, the Horse Selector calculates the estimated odds. BET on horses whose actual payoff (from the Tote Board or Morning Lines) is higher than payoff based on estimated odds.
- Using the above factors, the Horse Selector calculates the estimated odds. BET on any selected horse with an estimated payoff (based on Tote Board or Morning Lines) higher than calculated payoff (based on Horse Selector II).
- Source listing for the TRS-80™, TI-59, HP-67, HP-41, Apple and BASIC Computers.
- No computer or calculator necessary (although a calculator would be helpful for the simple division used to calculate estimated odds).

FREE **Dutching Tables** allows betting on 2 or more horses with a guaranteed profit.

NEWDOS/80

A **New** enhanced NEWDOS for TRS-80™ Model I for the 1980's

Apparat Inc., announces the most powerful Disk Operating System for the TRS-80™. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2. 1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2. 1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2. 1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:

- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disk drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOSREADY" or "READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST."
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful program chaining.
- Device hanging for routing to display and printer simultaneously.
- CDE function; simultaneous striking of the C, D and E keys will allow user to enter a mini-DOS to perform some DOS commands without disturbing the resident program.
- Upward compatible with NEWDOS 2. 1 and TRSDOS 2.3.
- Includes Superzap 3.0 and all Apparat 2.1 utilities.

..... **$149.00**

STOCK MARKET MONITOR

Galactic Software Ltd.

CASSETTE VERSION **$89.00**
DISK VERSION **$99.00**

1. The system is designed for the active "trader" not the "long term" investor, as the system is "technically" oriented.
2. For the TRS-80™ Model I, Level II, 16K or more. Available in both disk and tape versions.
3. Tracks user selected issues, in a technical system that reflects the issue's performance against the overall market.
4. Set up data is input by the user from the Standard and Poors stock guide or Value Line.
5. Daily issue data, "high", "low", "close" and "volume" are input from any newspaper containing this information.
6. Daily overall market, "volume" and "closing Dow" are also provided from a newspaper.
7. Volume and price changes of an issue, as they compare to volume an price changes of the overall market, are the basis of this system's analysis of the given issue.
8. Comparisons of the issue against itself are also done. This may allow the user to spot "unusual" activity on this issue.
9. Clear indications are given as to whether the issue is "out performing", "under performing" or "performing" with the market.
10. Complete video and printed output is provided.
11. This program is intended to be a guide to indications, and is not to be used as a sole recommendation to buy, sell or hold an issue. These decisions are the responsibility of the user and his brokerage.

COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

✓9 50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software (less $3 penalty for handling)

ADD $2.00 FOR SHIPPING IN UPS AREAS
ADD $3.00 FOR C.O.D. OR NON-UPS AREAS
ADD $4.00 OUTSIDE U.S.A., CANADA & MEXICO

MICRO COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

BUSINESS PAC 100

100 Ready-To-Run Business Programs

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software (less a $3 penalty for handling)

(ON CASSETTE OR DISKETTE).....Includes 110 Page Users Manual.....5 Cassettes (Or Diskettes)
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNU1	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
NAME		**DESCRIPTION**
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAPI	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ **CASSETTE VERSION $ 99.95**
☐ **DISKETTE VERSION $ 99.95**
☐ **MODEL II VERSION $149.95**

ADD $2.00 FOR SHIPPING IN UPS AREAS
ADD $3.00 FOR C.O.D. OR NON-UPS AREAS
ADD $4.00 OUTSIDE U.S.A, CANADA & MEXICO

MICRO COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

VISA

24 HOUR ORDER LINE
(914) 425-1535

H & E COMPUTRONICS INC. Announces ...MOD-II

MATHEMATICAL APPLICATIONS SERVICE™

- *Review of the latest Software for the TRS-80™ Model II Microcomputer*
- *Tips for running your Model II*
- *Latest announcements and updates*

The Newsletter For Owners Of The TRS-80™* MODEL II MICROCOMPUTER

SPECIAL CHARTER SUBSCRIPTION RATE $12.00 □ (For 12 issues)

MOD-II PROGRAMS CURRENTLY AVAILABLE

(1) **ELECTRIC PENCIL** (Michael Shrayer Software).....Complete word processor with extensive editing and printer formatting features.....$275 (STANDARD CP/M VERSION).....$300 (DIABLO, NEC OR QUME CP/M VERSION).....$325 (STANDARD TRSDOS VERSION).....$350 (DIABLO, NEC OR QUME TRSDOS VERSION).

(2) **GENERAL LEDGER, ACCOUNTS RECEIVABLE, ACCOUNTS PAYABLE, INVENTORY CONTROL AND PAYROLL** (Small Business Systems Group).....works under TRSDOS.....can be used one module at a time or as a coordinated system.....$225 per module.....$995 for the complete system.

(3) **GENERAL LEDGER, ACCOUNTS RECEIVABLE, ACCOUNTS PAYABLE, INVENTORY CONTROL AND PAYROLL** (Peachtree Software).... requires CP/M and MICROSOFT BASIC.....professional business systems.....turn key operation.....discounts available.....$1000 per module .....$5000 for the complete system.

(4) **SELECTOR III** (Micro-Ap).....complete data management system.....user defined fields and codes.....manages any list defined by the user..... includes additional modules for simplified inventory control, accounts receivable and accounts payable.....requires CBASIC-2 and CP/M .....$295.

(5) **GLECTOR** (Micro-Ap).....add on package to the SELECTOR III.....general ledger that allows the user to define a customized chart of accounts.....$250.

(6) **GSF** (Racet Computers).....Generalized Subroutine Facility.....a series of super fast machine language utilities that can be called from a BASIC program (no machine language knowledge required).....sorts 1000 items in under 5 seconds.....allows PEEK and POKE statements..... move data blocks.....compress and uncompress data.....works under TRSDOS.....$50.

(7) **DSM** (Racet Computes).....Disk Sort Merge.....sorts and merges large multiple diskette files on a 1 to 4 drive system.....NOT AN IN MEMORY SORT.....can actually alphabetize (or any other type or sort) 4 disk drives worth of data.....sorts one complete disk of information in 10 minutes.....information is provided to use DSM with the RS MAILING PROGRAM.....works under TRSDOS.....$150.

(8) **RSM** (Small Systems Software).....a machine language monitor and disassembler.....can be used to see and modify memory or disk sectors.... contains all the commands found on the Model-I version plus some additional commands for the MOD-II.....works under TRSDOS.....$39.95.

(9) **CP/M** (Lifeboat Associates).....an alternative operating system to TRSDOS that allows users to use hundreds of programs currently available to CP/M owners.....This is the only version of CP/M for the MOD-II that comes with an elementary CP/M guide written especially for MOD-II owners.....$170.

(10) **MICROSOFT BASIC** (Microsoft).....an enhanced version of the MICROSOFT BASIC found on TRSDOS.....works under CP/M.....adds commands such as chaining (allows the user to LOAD and RUN a new program without losing the variables currently in memory).....long variable length file records, WHILE/WEND and others.....can be used with the BASIC COMPILER to speed up programs (3-10 times faster execution).....$350.

(11) **CBASIC-2**.....a non-interactive BASIC used for many programs that run under CP/M.....requires CP/M.....allows user to make more efficient use of disk files.....eliminates the use of the most line number references.....required for programs such as SELECTOR and GLECTOR...$120.

(12) **MAILING ADDRESS** (Peachtree Software).....requires CP/M.....keeps track of name and address information and allows the selective printing of the information in the form of mailing lists or address labels.....unique key structure and formatting structure allows for a multitude of retrieval alternatives.....$790.

(13) **PROPERTY MANAGEMENT** (Peachtree Software).....requires CP/M.....keeps track of all financial records related to property management .....$1500.

(14) **FORTRAN-80** (Microsoft).....ANSI 66 (except for COMPLEX) plus many extensions.....requires CP/M.....$425.

(15) **H & E COMPUTRONICS, INC. SHARE-A-PROGRAM DISKETTE #1**.....works under TRSDOS.....a collection of programs written by MOD-II owners.....programs include data base management.....a word processor.....mail system.....mortgage calculations.....checkbook register..... and many others.....$8 (add $3 postage outside of the United States, Canada and Mexico).....FREE if you send us a diskette containing a program that can be added to the SHARE-A-PROGRAM DISKETTE.

(16) **MEMOREX OR WABASH CERTIFIED DISKETTES**.....$49.95 (per box of 10).

(17) **FLEXI-MATIC DISKETTE STORAGE TRAY**.....Stores 110 diskettes.....comes complete with index-dividers, dust cover, tilt plates and adjustable spacing.....$55.00.

(18) **WORD-STAR**.....The ultimate word processor.....a menu driven word processing system that can be used with any printer. All standard word processing commands are included.....plus many unique commands only found on WORD STAR.....requires CP/M.....$495.

(19) **MAIL LIST MERGE**.....An add on package that allows the user to send form letters (created on WORD-STAR) to any compiled mailing list (using any CP/M based MAIL program such as the PEACHTREE MAIL PROGRAM).....requires CP/M, WORD STAR and any CP/M based mail program.....$150.

(20) **EDITOR ASSEMBLER** from Galactic Software Ltd. is the first user oriented Editor Assembler for the MODEL II and was designed to utilize all the features of the MODEL II. It includes innovative features for ease of coding and debugging and complete documentation (over 120 pages) .....works under TRSDOS.....$229.

(21) **MAIL/FILE SYSTEM** from Galactic Software Ltd. stores 2,500 names per disk. No sorting time is required since the file is automatically sorted by first and last name plus Zip Code on input. Retrieve by any combination of 19 user codes. Supports an 11 digit alphanumerica Zip. Supports a message line. Comes complete with user-oriented documentation (100-page manual). Allows for company name and individual of a company and complete phone number (and extensions).....works under TRSDOS.....$199.

TRS-80™ IS A TRADEMARK OF TANDY CORPORATION

✓9

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

master charge

24 HOUR ORDER LINE

(914) 425-1535

VISA

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

✓Reader Service—see page 226

This nifty program screen displays your active variable and runs as a USR function to boot!

Variable Scroll

William L. Colsher
4328 Nutmeg Lane, Apt. 111
Lisle, IL 60532

If you have ever tried to develop a long BASIC program on a TRS-80, you have probably wondered at some point exactly what variable names you have already used. Even if you are fortunate enough to have a printer, it is often hard to be sure that a new variable is really new.

There are at least two ways around this problem. The simplest is to keep a list of the variables as you use them. Unfortunately, that requires more foresight than most of us use. Besides, that piece of paper can get lost all too easily.

The other solution is a program that displays the currently active variables on the screen any time. That program is the topic of this article.

Variable Locations

To begin, you need to know where (Table 1) and how (Table 2) BASIC stores its variables.

I decided to make this program a USR function.

Using it is quite simple. If you take a look at the *BASIC Reference Manual*, you'll find that in order to use the USR function, you have to POKE a couple of bytes in one of the reserved areas with the address of the USR routine.

This is ordinarily the case, but an assembler is a wonderful thing, especially one with an ORG statement. The first ORG in the program uses the address of the place we would have had to POKE. The next statement, a DEFW that contains the starting address of the program, is assembled at that location. (If you want to assemble this program at a different location, remember to change both the DEFW and the second ORG.)

Since the POKE is taken care of, all you have to do is protect an area of memory for the program by entering 32500 in response to "MEMORY SIZE?", loading the program with the SYSTEM command and returning to BASIC. Table 3 shows you how.

You can test the program now by DIMensioning a couple of variables in the immediate mode and assigning values to a couple of scalars. Invoke the DISP routine by typing PRINT USR(0) or X = USR(0). (X is only an example, any variable name can be used.)

The screen should clear and you will get a display of your scalars on one line or more: there are sixteen variables per line. Below that are the arrays.

A couple of improvements

Location	Contents
16633	The address of the start of the scalar area
16635	The address of the start of the array area
16637	The address of the start of free memory

Table 1: Part of the BASIC Reserved Area. All the areas are contiguous. Thus, the start of the second is also the end of the first, etc.

All variables are stored in essentially the same manner. The first three bytes always store the same information:

Type	2nd byte of name	1st byte of name

(Type may be one of the following: 2—Integer, 3—String, 4—Single Precision, 8—Double Precision.)
If a variable has a single character name, a zero is placed in the other location. After these three bytes, scalars have the following form:

LSB of value	MSB of value						
LSB of value	Next MSB of value	MSB of value	Exponent of value				
LSB of value	Next MSB of value	etc.	etc.	etc.	etc.	MSB of value	Exponent of value
Length of string	LSB of start address	MSB of start address					

(Variables are top to bottom: Integer, Single Precision, Double Precision, String.)
Following is the format of the array storage area. The first three bytes are the same as for scalars. This is followed by:

LSB of array size	MSB of array size	Number of dimensions

This is followed by pairs of bytes containing the dimension sizes. This is in turn followed by the actual values.

Table 2: Structure of BASIC Variable Storage.

come to mind after using this program for a little while. The variables are displayed in the order they were used. Scalars come before arrays, but that's built into the program.

It might be wise to sort the variables, so that when working with a large program it is easier to check a new variable name. A second but more difficult enhancement might add a cross-reference generator. This would display not only the variables, but also the line numbers in which each is used. ■

```
MEMORY SIZE? 32500 (carriage
return)
RADIO SHACK LEVEL II BASIC
READY
>SYSTEM (carriage return)


*?DISP (carriage return)
The asterisks will flash briefly.
*? (carriage return)
This returns to BASIC.
?SN ERROR

READY
>_
```

Table 3. Loading and Using DISP

```
00080 ;THIS PROGRAM PROVIDES AN EASY WAY TO OBTAIN
00090 ;A LIST OF CURRENTLY ACTIVE VARIABLES IN A
00100 ;LEVEL II BASIC PROGRAM.
00110 ;
00120 ;CALLING METHOD: POKE16527,126:POKE16526,244:PRINT USR(0)
00130 ;
00140 ;THE FOLLOWING ORG AND DEFW TAKE CARE OF THE POKE
00150 ;THAT WOULD ORDINARILY HAVE TO BE DONE TO TELL
00160 ;BASIC WHERE THE USR ROUTINE IS.
00170         ORG     16526
00180         DEFW    32500
00190 ;
00200 ;CLEAR SCREEN
00210         ORG     32500
00220         LD      HL,SCREEN
00230         LD      DE,SCREEN+1
00240         LD      BC,1023
00250         LD      A,' '
00260         LD      (HL),A
00270         LDIR
00280 ;DISPLAY SCALARS
00290 ;
00300 ;THERE ARE 2 GENERAL TYPES OF VARIABLES IN LEVEL II
00310 ;BASIC: SCALARS AND ARRAYS.  POINTERS TO THE AREAS
00320 ;BASIC USES TO STORE VARIABLES ARE TO BE FOUND IN
00330 ;ONE OF THE RESERVED AREAS OF RAM. SPECIFICALLY,
00340 ;THE CONTENTS OF THESE LOCATIONS ARE THE POINTERS:
00350 ;16633 -> START OF SCALAR AREA
00360 ;16635 -> START OF ARRAY AREA (END OF SCALARS)
00370 ;16637 -> START OF FREE MEMORY (END OF ARRAYS)
00380 ;
00390 ;THE FORMAT OF THE SCALAR AREA IS AS FOLLOWS:
00400 ;BYTE    CONTENTS
00410 ; 0      SCALAR TYPE - 2=INTEGER,3=STRING
00420 ;                      4=SINGLE,8=DOUBLE
00430 ; 1      2ND CHAR OF NAME
00440 ; 2      1ST CHAR OF NAME
00450 ; 3      IF STRING, LENGTH, ELSE LSB OF VALUE
00460 ;4-5     IF STRING -> START OF STRING, ELSE MORE VALUE
00470 ;6-10    VALUE
00480 ;
00490 ;FOR MORE INFORMATION SEE PAGES 8/8 AND 8/9 OF THE
00500 ;LEVEL II BASIC REF. MAN.
00510         LD      HL,(16633)      ;-> SCALAR AREA
00520         LD      DE,SCREEN       ;-> TOP OF SCREEN
00530 ;THIS CODE TAKES CARE OF THE CASW WHEN THERE ARE
00540 ;NO SCALARS ACTIVE IN THE BASIC PROGRAM
00550         PUSH    DE
00560         PUSH    HL
00570         LD      DE,(16635)
00580         SBC     HL,DE
00590         JP      Z,ARRAYS
00600         POP     HL
00610         POP     DE
00620 ;FOLLOWING CODE GETS THE SCALAR NAME
00630 ;A CALL TO 'STRCHK' PUTS IN A $ IF NECESSARY
00640 ;
00650 ;HL -> VARIOUS PLACES IN THE SCALAR AREA
00660 ;DE -> CURRENT SPOT ON THE SCREEN WHERE WE'LL
00670 ;       BE PUTTING SOMETHING.
00680 ;
00690 ;*NOTE* STRINGS WHICH HAVE NOT BEEN GIVEN A VALUE
00700 ;WILL *NOT APPEAR* ON THE LIST.
00710 SHOWSC  CALL    SHOWIT
00720         PUSH    DE
00730         LD      D,0
00740         LD      E,(HL)
00750         ADD     HL,DE
00760         INC     HL
00770         INC     HL
00780         INC     HL
00790         PUSH    HL
00800         LD      DE,(16635)
00810         SBC     HL,DE
00820         JR      Z,ARRAYS
00830         POP     HL
00840         POP     DE
00850         JR      SHOWSC
00860 ;FOLLOWING CODE DISPLAYS THE ARRAY NAMES.
00870 ;THE ARRAY AREA HAS THE FOLLOWING FORMAT:
00880 ;BYTE    CONTENTS
00890 ;0-2     SAME AS FOR SCALARS
00900 ;3-4     SIZE OF ARRAY
00910 ; 5      NUMBER OF DIMENSIONS
00920 ;6-?     VALUES
00930 ;
00940 ARRAYS  LD      DE,(16637)      ;END OF ARRAYS POINTER
00950         LD      HL,(16635)      ;START OF ARRAYS POINTER
00960         SBC     HL,DE           ;COMPARE WITH CURRENT LOC
00970         JP      Z,DONE          ;IF NO ARRAYS QUIT
00980         POP     HL              ;ELSE CLEAN UP STACK
00990         POP     DE              ;. .
01000         CALL    LINES           ;START A NEW LINE
01010 SHOWAR  CALL    SHOWIT
01020         PUSH    DE
01030         INC     HL
01040         INC     HL
01050         INC     HL
01060         LD      E,(HL)
01070         INC     HL
01080         LD      D,(HL)
01090         INC     HL
01100         ADD     HL,DE
01110         PUSH    HL
01120         LD      DE,(16637)
01130         SBC     HL,DE
01140         JP      Z,DONE
01150         POP     HL
01160         POP     DE
01170         JP      SHOWAR
01180 ;AND NOW WE CLEAN UP AND RETURN TO BASIC
01190 DONE    POP     HL
01200         POP     DE
01210         CALL    LINES   ;POINT TO NEXT LINE
01220         LD      (16416),DE      ;AND FIX R.S. CURSOR
01230         RET
01240 SCREEN  EQU     3C00H
01250 ;THIS CODE CHECKS THE TYPE FIELD AND IF IT IS
01260 ;A 3 WE KNOW THAT THE VARIABLE IS A STRING. IN
01270 ;THAT CASE WE INSERT A $ SO THE USER WILL KNOW
01280 ;TOO.
01290 ;
01300 ;REGISTER B IS USED AS A FLAG TO TELL IF WE HAD
01310 ;TO INCREMENT THE DE PAIR TO ACCOMODATE A 2 CHARACTER
01320 ;VARIABLE NAME.
01330 ;
01340 ;AT THIS TIME HL POINTS TO BYTE ZERO OF THE CURRENT
01350 ;VARIABLE STORAGE AREA.
01360 STRCHK  LD      B,0
01370         CP      ' '
01380         JP      Z,DONT
01390         LD      B,255
01400         INC     DE
01410 DONT    LD      A,(HL)
01420         CP      3
01430         JP      NZ,SKIPIT
01440         LD      A,'$'
01450         LD      (DE),A
01460 SKIPIT  LD      A,B
01470         CP      0
01480         RET     Z
01490         DEC     DE
01500         RET
01510 ;THE FOLLOWING CODE MOVES THE SCREEN POINTER TO A
01520 ;NEW LINE TO SEPERATE THE SCALARS FROM THE ARRAYS
01530 ;VISUALLY.  IT IS ALSO CALLED JUST BEFORE RETURNING
01540 ;TO BASIC TO SET THE BASIC CURSOR TO SUITABLE
01550 ;LOCATION.
01560 ;
01570 ;AT THIS TIME REGISTER DE POINTS TO WHAT WOULD
01580 ;BE THE NEXT SCREEN LOCATION TO PUT A VARIABLE NAME
01590 ;IN.  SINCE THE SCREEN LINES ARE ON 64 BYTE BOUNDRIES
01600 ;THE BRUTE FORCE METHOD BELOW WORKS NICELY.
01610 LINES   LD      A,E
01620         CP      40H
01630         JP      M,DO40
01640         CP      80H
01650         JP      M,DO80
01660         CP      0C0H
01670         JP      M,DOC0
01680         LD      E,0
01690         INC     D
01700         RET
01710 DO40    LD      E,40H
01720         RET
01730 DO80    LD      E,80H
01740         RET
01750 DOC0    LD      E,0C0H
01760         RET
01770 ;FOLLOWING CODE HANDLES THE DISPLAY OF VARIABLE
01780 ;NAMES FOR BOTH SCALARS AND ARRAYS.
01790 ;BOTH TYPES OF VARIABLES HAVE THE SAME FORMAT IN
01800 ;THE FIRST THREE BYTES: TYPE, CHAR2, CHAR1.
01810 ;NOTE THAT IF A VARIABLE HAS ONLY A SINGLE
01820 ;CHARACTER NAME CHAR2 CONTAINS A *ZERO* NOT A
01830 SHOWIT  INC     HL
01840         INC     HL
01850         LD      A,(HL)
01860         LD      (DE),A
01870         INC     DE
01880         DEC     HL
01890         LD      A,(HL)
01900         CP      0
01910         JR      NZ,OK
01920         LD      A,' '
01930 OK      LD      (DE),A
01940         DEC     HL
01950         CALL    STRCHK
01960 ;POINT DE TO NEXT SCREEN LOCATION. I WASTE AN EXTRA
01970 ;BYTE TO MAKE THE LINES COME OUT EVEN ALL THE TIME.
01980 ;I.E. ALWAYS 16 VARICBLES PER LINE.
01990         INC     DE
02000         INC     DE
02010         INC     DE
02020         RET
02030         END
```

Program Listing

PROGRAMMING TOOLS FOR YOUR TRS-80

INSIDE LEVEL II

The Programmers Guide to the TRS-80 ROMS

INSIDE LEVEL II is a comprehensive reference guide to the Level II ROMs which allows the machine language or Basic programmer to easily utilize the sophisticated routines they contain. Concisely explains set-ups, calling sequences, and variable passage for number conversion, arithmetic operations, and mathematical functions, as well as keyboard, tape, and video routines. Part II presents an entirely new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer including tape formats, RAM useage, relocation of Basic programs, USR call expansion, creating SYSTEM tapes of your own programs, interfacing of Basic variables directly with machine code, a method of greatly increasing the speed at which data elements are stored on tape, and special precautions for disk systems. INSIDE LEVEL II is a clearly organized reference manual. It is fully typeset and packed with nothing but useful information. It does not contain questions and answers, ROM dumps, or cartoons. **INSIDE LEVEL II..... 15.95**

TELECOMMUNICATIONS PROGRAM

This program allows reliable high speed file transfers between two disk-based computers over modems or direct wire. It is menu driven and extremely simple to use. Functions include real-time terminal mode, save RAM buffer on disk, transmit disk file, receive binary files, examine and modify UART parameters, program 8 custom log-on messages, automatic 16-bit checksum verification of accurate transmission and reception, and many more user conveniences. Supports line printers and lowercase characters. With this program you will no longer need to convert machine language programs to ASCII for transmission, and you will know immediately if the transmission was accurate. **TELCOM.....$29.95**

PROGRAM INDEX FOR DISK BASIC

Assemble an alphabetized index of your entire program library from disk directories. Program names and free space are read automatically (need not be typed in) and may be alphabetized with a fast Shell/Metzner sort by disk or program. The list may also be searched for any disk, program, or extension; disks or programs added or deleted; and the whole list or any part sent to the printer. Finally, the list itself may be stored on disk for future access and update. "The best thing since sliced bread" (January issue of '80 Microcomputing). One drive and 32K required. **INDEX.....$19.95**

SINGLE STEP THROUGH RAM OR ROM

STEP80 allows you to step through any Basic or machine language program one instruction at a time, and see the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The top 14 lines of the video screen are left unaltered so that the "target program" may perform its display functions unobstructed. STEP80 will follow program flow right into the ROMs, and is an invaluable aid in learning how the ROM routines function. Commands include step (trace), disassemble, run in step mode at variable step rate, display or alter memory or CPU registers, jump to memory location, execute a CALL, set breakpoints in RAM or ROM, and relocate to any page in RAM. The display may also be routed to your line printer through the device control block so custom print drivers are automatically supported. **STEP80.....$16.95**

4 SPEED OPTIONS FOR YOUR TRS-80!

The SK-2 is the most versatile clock modification available for the TRS-80. Speeds may be switched between normal, an increase of 50%, or a 50% reduction; selectable at any time without interrupting execution or crashing the program. Instructions are also given for a 100% increase to 3.54 MHz, though the TRS-80 is not reliable at this speed. The SK-2 may be configured by the user to change speed with a toggle switch or on software command. It will automatically return to normal speed any time a disk is active, requires no change to the operating system, and has provisions for adding an LED to indicate when the computer is not at normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with socketed IC's and illustrated instructions. **SK-2.....$24.95**

RAM SPOOLER AND PRINT FORMATTER

This program is a full feature print formatting package featuring user defineable line and page length (with line feeds inserted between words or after punctuation), screen dump, and printer pause control. The serial version allows baud rate selection from the keyboard. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. Please specify PARALLEL or SERIAL (RS-232 interface) version. **SPOOLER.....$16.95**

DUPLICATE SYSTEM TAPES WITH CLONE

Make duplicate copies of ANY tape written for Level II. They may be SYSTEM tapes (continuous or not) or data lists. The file name, load address, entry point, and every byte (in ASCII format) are displayed on the video screen. **CLONE.....$16.95**

MACHINE CODE FAST FOURIER TRANSFORM

This complete package includes 3 versions of the machine language FFTASM routine assembled for 16, 32, and 48K machines, a short sample Basic program to access them, a 10K Basic program which includes sophisticated interactive graphing and data manipulation, and a manual of instructions and examples. The machine language subroutines use variables defined by a supporting Basic program to make data entry and retrieval extremely fast and easy for custom implementation. They perform 20 to 40 times faster than their Basic equivalent (256 points in 12.5 seconds), and require less than 1550 bytes of memory. **FFTASM.....49.95**

FOR THE MODEL II

LYNC

from Midnight Software

High level data communication for the Model II with CP/M. LYNC will send and receive any file with automatic error checking and retries. Either end may initiate file transfers, and multiple files may be sent with wildcard filenames. Remote or local directories may be called from within the program. Allows full protocol, non-protocol, and real-time conversation modes. May be used over phone lines at 300 baud or direct to another computer at up to 9600 baud. Also available for other CP/M computers. **LYNC.....$95.00**

MUMFORD MICRO SYSTEMS

✓144

ORDERING: Complete satisfaction is guaranteed or a full refund will be made. All Model I programs are shipped on cassette unless $5 is included for a formatted (no system) disk. Include $1 postage and handling. California residents add 6% sales tax. Visa, Mastercharge and COD orders accepted.

Box 435-E Summerland, California 93067 (805) 969-4557

Frustrated by the limitations of INPUT command? Read this.

Input with Insight

Jack Decker
1804 West 18th Street, Lot 155
Sault Ste. Marie, MI 49783

Most TRS-80 users have been frustrated by the limitations of the INPUT command in BASIC.

The command won't accept commas in the input string unless quotation marks are also used, and it won't select an appropriate prompt character in place of the question mark. (How many times have you seen PRESS ENTER TO CONTINUE? in a program?) It also has difficulty formatting the input on the screen, since hitting the enter key always advances the cursor to the beginning of the next line.

Construct a String

The solution is to construct a string using the INKEY$ function. The subroutine in Listing 1 does just this. It's the shortest subroutine I've found that will overcome the limitations, and yet behave like an INPUT command in regard to the use of the backspace and shifted backspace keys.

No automatic line feed is needed after the enter key is pressed, but if you want one you can insert a PRINT: statement just before the RETURN. In the interest of conserving memory line 60010 is heavily packed.

To use the subroutine in place of an INPUT statement, simply GOSUB 60000. On return, string variable B$ will contain the input. You use variable A$ within the subroutine to hold individual keyboard strokes as they are entered. No other variables are used in this routine.

For those who may wish to customize this routine for their own applications, here is a brief explanation of what is happening in the subroutine:

In line 60000, B$ is set to the null string (a string variable with no characters in it). PRINT CHR$(14); turns on the cursor.

Lines 60010 and 60020 form a loop. When the program falls through from line 60010, line 60020 catches it and sends it right back where it came from.

Line 60010 first tests to see that a character has been input. If one has (A$>""), it then tests to see if that character was the enter key. If so, it turns off the cursor (PRINT CHR$(15);) and returns.

If the character was not the enter key, it is tested to see that it is not a control character (IF A$>CHR$(31)). This is to keep anyone from lousing up the whole screen display by accidentally hitting the wrong key.

Assuming the character passes this test, you test the string variable to make sure it would not contain over 255 characters if this character were added to it (LEN(B$)<255); this causes an LS ERROR.

Here is your opportunity to limit the length of the input. If, for example, you want to prevent any input over 40 characters long, change the 255 to 40. You could even change the 255 to a numeric variable, such as X, and then use a statement of the form: X = 40:GOSUB 60000.

Varying the value of X allows a different maximum line length each time the subroutine is called.

Character Input

Going back to the character that was input, assume that it has an ASCII value greater than 31 and will not cause B$ to ex-

```
60000 B$="":PRINTCHR$(14);
60010 A$=INKEY$:IFA$>""IFA$=CHR$(13)THENPRINTCHR$(15);:
RETURNELSEIFA$>CHR$(31)ANDLEN(B$)<255THENPRINTA$;:B$=B$
+A$ELSEIFB$>""IFA$=CHR$(8)THENPRINTA$;:B$=LEFT$(B$,LEN(
B$)-1)ELSEIFA$=CHR$(24)THENPRINTSTRING$(LEN(B$),8);:B$=
""
60020 GOTO60010
```

Program Listing 1.

```
60000 B$="":Z=0
60010 Z=NOTZ:PRINTCHR$(15+Z);:FORX=1TO12:A$=INKEY$:IFA$
>""IFA$=CHR$(13)THENPRINTCHR$(15);:RETURNELSEIFA$>CHR$(
31)ANDLEN(B$)<255THENPRINTA$;:B$=B$+A$ELSEIFB$>""IFA$=C
HR$(8)THENPRINTA$;:B$=LEFT$(B$,LEN(B$)-1)ELSEIFA$=CHR$(
24)THENPRINTSTRING$(LEN(B$),8);:B$=""
60020 NEXT:GOTO60010
```

Program Listing 2.

```
5 CLEAR500
10 PRINT"NAME:"TAB(12);:Y=32:GOSUB60000:PRINT:N$=B$
20 PRINT"ADDRESS:"TAB(12);:Y=32:GOSUB60000:PRINT:M$=B$
30 PRINT"CITY:"TAB(12);:Y=22:GOSUB60000:C$=B$
40 PRINTTAB(37)"STATE: ";:Y=2:GOSUB60000:S$=B$
50 PRINTTAB(49)"ZIP: ";:Y=5
60 GOSUB60000:IFLEN(B$)<5THENPRINTSTRING$(LEN(B$),8);:G
OTO60
70 PRINT:Z$=B$
100 PRINT:PRINTN$:PRINTM$:PRINTC$", "S$"  "Z$:PRINT:GOT
O10
```

Program Listing 3.

ceed maximum length if added to it. The character is then added to B$ and printed on the screen.

Suppose it fails these tests? You still want to check for a valid backspace or shifted backspace code. But first you must test string variable B$ to make sure that it has one or more characters in it.

If B$ does indeed contain one or more characters, the routine tests to see if A$ is a backspace: (IF A$ = CHR$(8)). If so, it prints it on the screen and deletes one character from B$: (B$ = LEFT$(B$,LEN(B$)-1.

If A$ is a shifted backspace: (IF A$ = CHR$(24)), then as many backspace characters are printed as there are characters in B$, and B$ is set to the null string. Should the input character fail all tests, it is ignored.

If you prefer a blinking cursor, you can have that and still have only three lines in your subroutine. Listing 2 shows how. Line 60010 is very packed, but you save much space.

Variables X and Z are used in this subroutine, along with A$ and B$. X controls the rate of blink. Z will always equal either zero or −1, since NOT 0 = −1 and NOT −1 = 0. This means that CHR$(15 + Z) is always either CHR$(14), which turns on the cursor, or CHR$(15), which turns it off.

Using the Subroutine

Listing 3 shows one way to use this subroutine. Try typing this along with the subroutine of your choice. Experiment with it awhile, and then try replacing the 255 in line 60010 with variable Y. Note that you cannot enter more characters than the proper amount.

Entries for city, state and zip code are all on the same line of the video display. One type of error handling is shown in line 60; if the zip code entered is less than five characters, the program deletes the errant entry and forces you to reenter. Of course, you would want a more sophisticated error-checking routine in any serious program, but this demonstration program does show the capabilities of this subroutine.

One thing must be said about this subroutine—it violates all rules of stylish BASIC. If you like nice neat listings, then this subroutine is not for you.

For this I make no apologies, since I don't know of any way to make this routine list neatly without using a lot more memory. Part of its virtue is that it's short, making heavy use of the IF . . . THEN . . . ELSE syntax, and any attempt to make it more stylish destroys that virtue. ■

✓Reader Service—see page 226

All you need to read/write on both sides of a minifloppy is a paper punch and guts.

Punch Out Your Disks

Richard Taylor
100 Manhattan Ave. 1809
Union City, NJ 07087

For the price of a standard paper punch you can double your present disk storage. Just follow the simple steps detailed in this article and you can be reading and writing on both sides of your disks. I have punched-out over 100 disks and only two of them have had defective second sides.

To get started you will need a pencil, a paper punch that catches its own punches, a tracing of a disk and a smooth piece of paper. The tracing of the disk (which we shall call the 'templet') can be made by Xeroxing a disk, cutting apart an unusable disk, or by making a tracing.

Try to use a stiff piece of paper or glue the copy to a piece of cardboard. Cut out the center hole, the oblong area below it, the write protect notch on the upper right edge and the small hole near the center hole. The smooth piece of paper can be the backing from a peel-away label or something similar. The templet shown in the photographs was made from a Xerox. It happens to have two holes punched. This is just a convenience and is not needed to do the job.

The Second Hole

The only thing that prevents a Radio Shack disk drive from writing to the second side of your disks is that it needs a second small hole near the center so that it can find the sectors correctly. If you rotate your disk in its sleeve and watch the small hole, you will see an even smaller hole right in the disk. Soft sectored disks have only one of these and the disk drive uses a light to "see" when this tiny hole passes by.

Our job is to punch a second hole in the sleeve so that when the disk is flipped over it will have a hole that allows the drive to see the tiny hole in the disk. The placement of the second hole does not have to be perfect. As long as the tiny hole can be seen through the new holes in the sleeve, everything will run correctly.

STEP 1: With the label of the disk in the upper left hand corner, place the templet on the disk so that the small hole is positioned by the lower left side of the center hole. Line up all reference points. Using a pencil, trace the new small hole on the disk. Also trace the notch on the upper left edge (Photo 2).

STEP 2: Take the strip of smooth paper and insert it between the sleeve and the disk (Photo 3).

STEP 3: Using your thumb, make room for the punch by lifting the sleeve near the center hole (Photo 4).

STEP 4: Insert the punch and line it up with the traced hole. Punch the hole (Photo 5).

STEP 5: Insert your finger where the punch was and check to see if the linear has been completely removed. In most cases it will not be. With your finger, push it up through the hole and tear it off (Photo 6).

STEP 6: Repeat steps 1 through 5 on the second side of the disk.

STEP 7: Punch the new notch near the bottom of the label. (Photo 7).

That's all there is to it. Photo 8 shows you what your new disk should look like. Labels can be placed in the upper left corner with no problems. Any problems

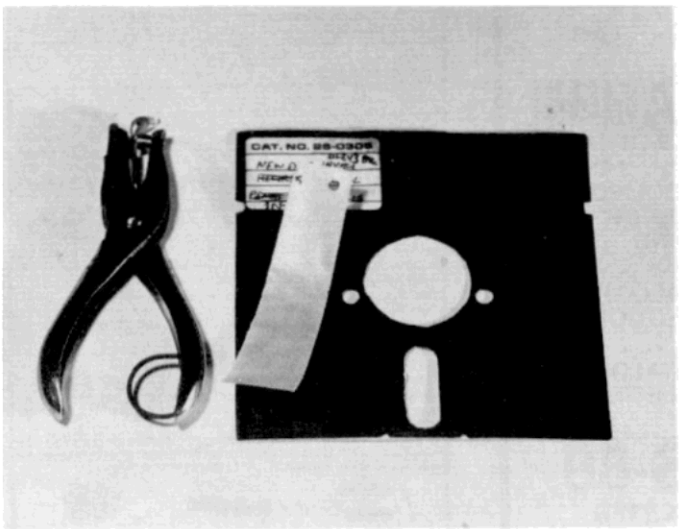


Photo 1.

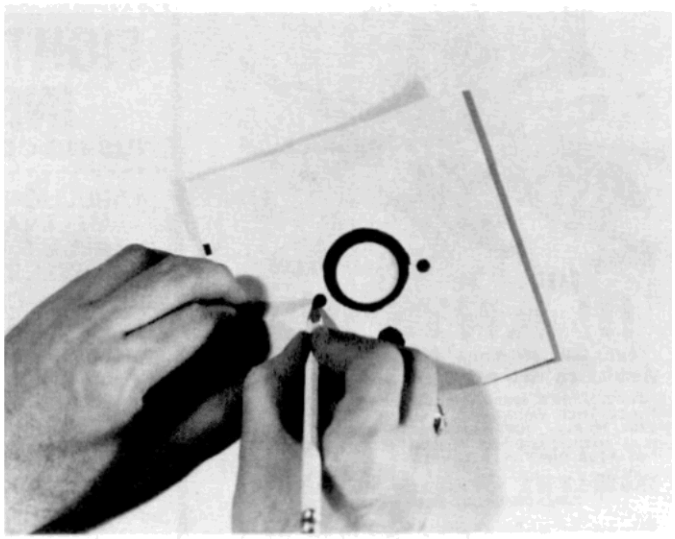


Photo 2.

SNAPP, INC.

Number I in software for the Model II

MODEL II EXTENDED BASIC

A family of enhancements to the Model II BASIC interpreter. Part of the package originated with the best of APPARAT, INC.'s thoughts in implementing NEWDOS BASIC. The system is written entirely in machine language for SUPER FAST execution. The extensions are fully integrated into Model II BASIC, and require NO user memory, and NO user disk space. The package is made up of the following five modules, each of which may be purchased separately:

XBASIC - Six single keystroke commands to list the first, last, previous, next, or current program line, or to edit the current line. Ten single character abbreviations for frequently used commands: AUTO, CLS, DELETE, EDIT, KILL, LIST, MERGE, NEW, LLIST, and SYSTEM. $25

XREF - A powerful cross-reference facility with output to display and/or printer. Trace a variable through the code. Determine easily if a variable is in use. $40

XDUMP - Permits the programmer to display and/or print the value of any or all program variables. Identifies the variable type for all variables. Each element of any array is listed separately. $40.

XRENUM - An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered, supports relocation of renumbered blocks of code, and supports duplication of blocks of code. $40

XFIND - Permits quick and easy location of specified strings or keywords within the program text. $30

SAVE - on the purchase of the entire package. $140

SKRUNCH

A SUPER FAST TRSDOS UTILITY. Compresses your BASIC programs to an absolute minimum. Typically saves 30-40% space, even for programs without REM statements! Also results in 7-10% improvement in execution speed. $35

VERBATIM DISKETTES

#4443 (MD525-01) — Suitable for Model I, or Model III, including double density drives.

Quantity (Boxes)	Price Per Box	Quantity (Boxes)	Price Per Box
1	$27.00	6	$25.75
2	26.75	7	25.50
3	26.50	8	25.25
4	26.25	9	25.00
5	26.00	10	24.50
		20+	24.00

#4495 (FD34-8000) — This is the preferred diskette for the Model II. Double density certified.

Quantity (Boxes)	Price Per Box	Quantity (Boxes)	Price Per Box
1	$37.50	6	$36.25
2	37.25	7	36.00
3	37.00	8	35.75
4	36.75	9	35.50
5	36.50	10	35.00
		20+	34.50

#3718 (FD34-1000) — This single density diskette will work on the Model II, but is NOT RECOMMENDED for critical applications.

Quantity (Boxes)	Price Per Box	Quantity (Boxes)	Price Per Box
1	$29.00	6	$27.75
2	28.75	7	27.50
3	28.50	8	27.25
4	28.25	9	27.00
5	28.00	10	26.50
		20+	26.00

We can supply any VERBATIM items at similar savings. Call or write with your requirements.

$5 shipping charge on MEDIA orders, regardless of size of order. This charge waived IF software is purchased on the same order.

FRIEND

FOUR NEW TRSDOS COMMANDS!

SHOW — A much better multi-disk directory display. Let's you see only those files you want, and includes date of last update.

MOVE — A much better file copying command. Copy/Move whole groups of files, renaming them at the same time, if desired, with just 1 command!

ERASE — Better than KILL, better than PURGE.

PRINT — Print BASIC programs from disk, whether saved in ASCII or compressed.

All 4 DOS commands allow fast processing of one, or complete groups of files, based on generic naming and wild card specifications. Enhanced functions too numerous to fully describe here.

EXAMPLES:

SHOW PAY*/BAS:*

Directory display of all '/BAS' files on all diskettes which begin with 'PAY.'

MOVE PAY*/BAS:1 TO =/OLD:3

Save current versions of payroll programs to drive 3, changing extension to '/OLD.'

MOVE OLD*/* TO NEW=/=:1

Copy all files on drive 0 which begin with 'OLD,' regardless of extension, to drive 1, changing the first 3 letters of the filename to 'NEW,' but retaining the same file extension.

Save time!

Reduce frustration!

Eliminate ERROR 33! $75

SPOOLER — Model I and Model II

Our workhorse! This package, available for Model I, in the TRSDOS/NEWDOS or NEWDOS 80 versions, or for the Model II, greatly enhances system performance when running typical business applications. Many applications have been benchmarked to run nearly TWICE AS FAST with the SPOOLER installed. Installs in minutes, and no changes are required to your programs. Preferred Model II versions require NO user memory. Optional features for the Model II version only: Serial printer support, and DISK SPOOLING support. The DISK SPOOLING support is particularly recommended for word processing applications. $100

SERIAL PRINTER OPTION $50

DISK SPOOLING OPTION $50

ITOII

A helping hand when converting BASIC programs from the Model I to the Model II. Automatically adjusts PRINT @, and PRINT USING to compensate for differences in the language. Advises you where adjustments are necessary for PEEK, POKE, etc. $25

BUGZAP

A powerful utility oriented toward the machine language programmer. Display/Modify/Print/Memory/Disk sectors. Use this to help you learn more about the internals of the Model II. $50

CALL TOLL FREE NOW

1 - 800 - 543-4628

Ohio residents call collect

(513) 891-4496

Snapp, Inc. ↙232

8160 Corporate Park Dr.

Cincinnati, Ohio 45242

Most products will soon be available for the Model I. CALL FOR DETAILS!

VISA

HOSTII / TERMII

Allows 'remote control' of a Model II from another Model II, or any ASCII terminal. If terminal is a Model II, accurate screen positioning (PRINT @) is fully supported! Requires NO user memory! This system is designed to provide software support to our customer locations without ever leaving the office. $50

MASTER / SLAVE

This software package was designed to support the transferring of files from one Model II to another, via direct connection or modem/phone line connection. ALL kinds of files, and baud rates up to 9600 are fully supported. Transfer files in either direction, even with the SLAVE Model II UNATTENDED! $150

ECHO

Causes all information going to the video display to also be routed to the printer. Instant hard copy. Can be turned on and off at will from within your program. Requires NO user memory. $35

ROUTE

Causes LPRINT data to be sent to the video screen! A great help in writing and debugging programs when no printer is available, you have a slow printer, or you are just in a hurry. Can be turned on and off from within your BASIC program. Requires NO user memory. $25

SCREEN

Supports the copying of the full video screen to the printer. Can be invoked by the operator with a keystroke, or from your program with a USR call. Requires NO user memory $25

SAVE

Retrieve the resident BASIC program following an accidental re-boot, an accidental SYSTEM, or a system crash. DON'T BE WITHOUT THIS ONE, YOU NEVER KNOW WHEN YOU WILL NEED IT! $35

SBASIC — Model I and Model II

Program in a high-level, full structured BASIC! The BEST of the BASIC pre-processors. PERFORM named subroutines. CONDITIONAL case structures. WHILE loops. UNTIL loops. And much more. Forget about line numbers. Model II version is compiled, and SUPER FAST. From Ultimate Computer Systems.

Model I $50

Model II $75

DOSFIX

A collection of patches to TRSDOS and BASIC to enhance their usability and function. Includes our well-known BREAK7E patches to keep the break key from being used accidentally. FREE WITH ANY MODEL II SOFTWARE PURCHASE.

TERMS OF SALE:

Credit card customers, add 3%. C.O.D. customers add $4. Ohio residents add 4½% sales tax. Shipments normally made the same day we receive your order.

OUR GUARANTEE:

If your diskette arrives damaged, we will replace it without charge. If you ever accidentally damage it, we will replace it for a $10 handling charge. For a period of one year, we will provide you with any enhancements or updates for a $10 handling charge. For a period of one year, if errors are discovered in the programs, they will be corrected without charge. In the event we cannot correct an error, you may return the program material for a refund.

THE FINE PRINT

TRS-80 is a trademark of the Radio Shack division of Tandy Corporation.

NEWDOS and NEWDOS/80 are trademarks of Apparat, Inc.

↙Reader Service—see page 226

with the new side will show up immediately just as they would with a new disk. There is no need to treat this new style any differently. All of my disks are double-sided and while I was unsure at first, I now have no fear of using the second side for the most important programs and data. In the early days there were problems involving bulk erasing. Now we have 2.2, 2.3, 3.0 and NEWDOS. All of these operating systems will backup over a disk that contains data without requiring bulk erasing.■

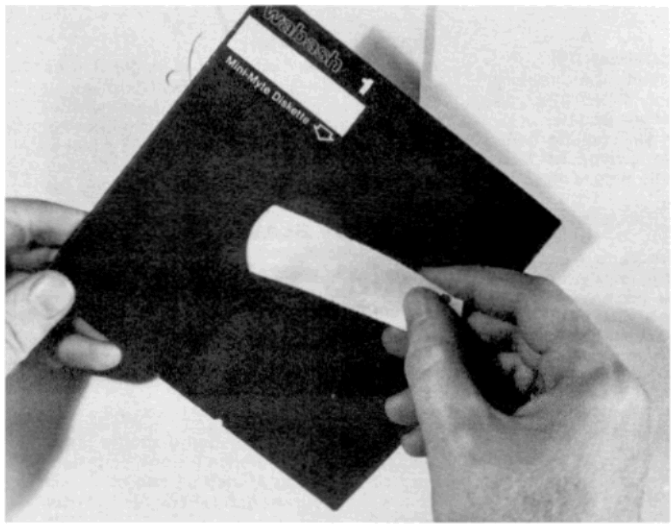


Photo 3.

Photo 6.

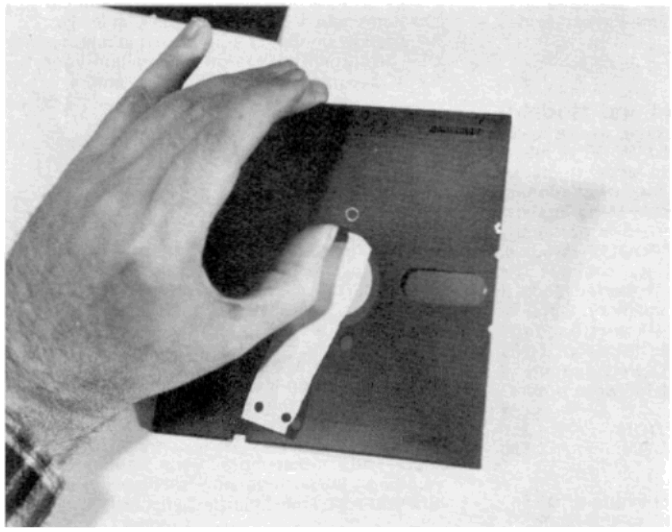


Photo 4.

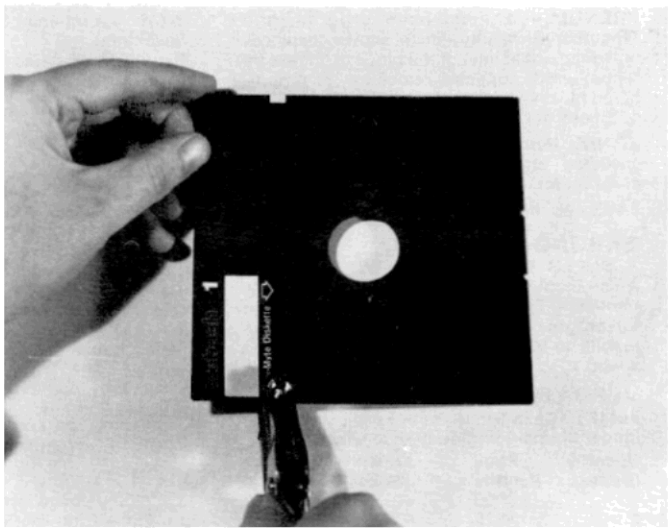


Photo 7.

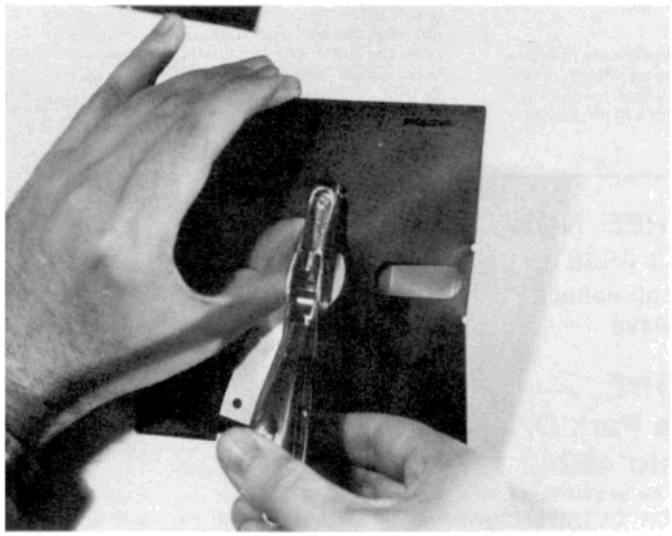


Photo 5.

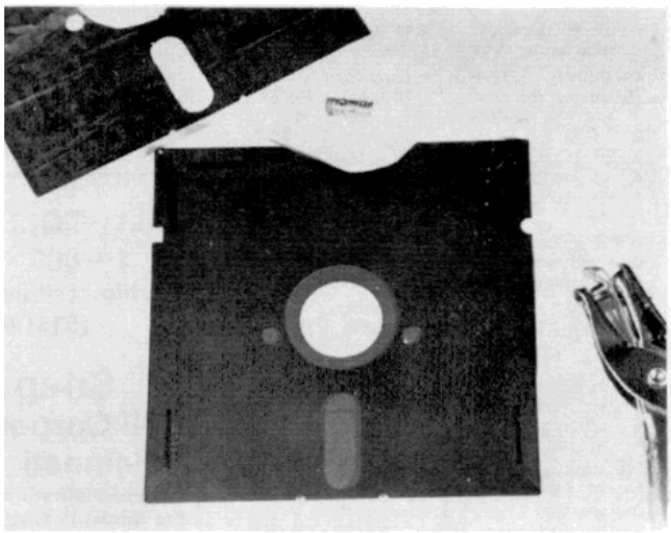


Photo 8.

I'M A BELIEVER !!

"I Love it !!. . .It's really a incredible O/S. It' just great! Now I see why people who have seen it say they are now believers. I know I am."

LANCE MICKLUS

1) Large (8") drive support.
2) Double Sided drive support.
3) Double Density drive support.
4) 80 Track drive support.
*NOTE all above drives may be mixed on any one system and can be configured at Sysgen time or during any Backup!
5) Winchester technology fixed drive support.
6) Supports any combination of the above drives up to a max. of 8 drives.
7) Supports double-speed processor clock modifications. (Archbold for example)
8) FASTER! ——— Improved overlay structure using ISAM accessing techniques improves loading times by up to 1400%.
9) General purpose output spoolers of a true, symbiont design provide simultaneous output and program execution without any user intervention.
10) Keyboard Type-Ahead feature permits you to enter keystrokes before your programs need them.
11) User definable keys, all 26 letters.
12) Built in Graphic string packer lets you enter graphic symbols into a BASIC program from the keyboard through the use of the (Clear) key. The (Clear) key is simply held down (just like the (Shift) keys) during other keystrokes and viola...graphics!
13) Dated files. —— All files are accompanied by the date of their last modification (creation or write).
14) Marked files. —— All files are accompanied by a 'mark' if they have been modified since they were last backed up. This permits the BACKUP utility to copy only those files which have actually been updated since a previous backup.
15) File transfer by class. Allows transferring of all files of a similar directory classification such as /CMD, /BAS, /PCL, etc.

VTOS 4.Ø

VTOS 4.Ø
Operating System Diskette with Operator's Guide
$99.95

VTOS 4.Ø
Master Reference Manual
$29.95

VTOS 4.Ø
Combination - 4.0 disk, Operator's Guide, and Master Reference Manual
$125.00

16) Built-in SYSTEM command contains lower case display driver, screen print, break key disable, blinking cursor, disk drive stepping rate and motor-on delay modifications, and more.
17) User may SYSGEN a custom VTOS system configuration containing special I/O drivers, device LINKing and ROUTEing, SPOOLing and DEBUG tasks, etc. which will be automatically loaded during the BOOT process without requiring a more lengthy AUTO and CHAIN procedure.
18) Non-BREAKable AUTO and CHAIN commands.
19) Wild-card DIRectory. Permits you to locate all files of a certain classification such as '/BAS'. Uniformly indicates file size in K (1024 bytes) regardless of drive type. "DIR D" would give you all your files that start with "D".
20) Dynamic file name defaults in APPEND, COPY, and RENAME commands allow you to specify only minimal information about file names.
21) COPY and APPEND commands execute up to 300% faster.
22) ALLOCate command for pre-allocation and non-releasibility of file space. File space will never shrink if this option used.
23) MEMORY command for directly setting upper memory limit.
24) Variable Length file support is incorporated which automatically blocks short user data records both within a sector and across sector boundaries thereby taking maximum advantage of disk file space.
25) No security disk needed to make backups or to run the system!
26) Though many O/S bear his design and code VTOS 4.Ø is the only Fully Aproved Operating System by Randy Cook! And it is FANTASTIC!
27) Endorsed by Scott Adams and Lance Micklus!

VTOS and VTOS 4.0 are registered trademarks of VIRTUAL TECHNOLOGY, INC. - Dallas, Texas 75234

Available from the following distributors or your local computer store. ***DEALER INQUIRIES INVITED.***

5% Discount Just For Mentioning This Ad. (Valid month of this publication ONLY)

QS QUALITY SOFTWARE DISTRIBUTORS
11234 Park Central Pl Suite C
Dallas Texas 75230
(214) 692-1055
Micronet - 70130,203
SOURCE - TCC293

Adventure INTERNATIONAL
ADVENTURE INTERNATIONAL
Box 3435, Longwood, Fla. 32750
(305) 862-6917 - Voice
after 8:00 - same number
as FORUM 80. (SOURCE - TCC957)

SMALL BUSINESS SYSTEMS GROUP
6 Carlisle Rd.
Westford, Mass 01886
(617) 692-3800 - Voice
(617) 692-3973 - FORUM 80
Micronet - 70310,236

Prices subject to change without notice.

All relationships have their ups and downs, this one is no exception.

Interfacing the NEC Spinwriter

James D. Kunzman
2221B Pennsylvania Avenue
Homestead AFB, FL 33039

Have you ever purchased a major peripheral for your TRS-80 microcomputer only to find that no interface information was provided?

This happened to me when I purchased my NEC Spinwriter.

After a fruitless call to the distributor, I was referred to an NEC Field Engineer who provided wiring instructions for running at 300 baud. He also suggested that I buy a $3.00 Product Description Manual explaining the wiring requirements. This manual clearly states the interface requirements for the RS-232-C port. If I had received it with the printer, I could have saved hours of grief.

While trying to connect this printer, I have discovered a simple technique to operate it at 1200 baud.

Hardware Problems

Using the NEC field engineer's instructions and the software driver published in Radio Shack's RS-232-C manual, the printer soon came to life—at least for a while.

I wired the Spinwriter according to the diagram in Fig. 1, minus the connection from pin 6 to pin 19, which is not required at 300 baud.

Suddenly the printer started printing garbage and made sporadic carriage movements. Eventually, it stopped printing completely.

Convinced that the printer was not at fault, I sent the expansion interface and RS-232-C to the Radio Shack repair center. The repair center merely reseated the RS-232-C board in the interface. This appeared to solve the problem.

After a week, however, it failed again, so I returned it to Radio Shack. Finally, after failing a *third* time, I took the board to the repair center myself to confront the repairman. We determined that the RS-232-C board was warping from heat, which caused it to fail.

"After failing a third *time I took the board . . . to confront the repairman."*

Since my board was slightly warped, the repair center cheerfully replaced the defective RS-232-C board with a new one. The baud rate generator runs rather warmly, so to keep this replacement board from warping I drilled 3/8-inch holes in the RS-232-C compartment door for ventilation. To further reduce heat problems I also removed the power packs from the expansion interface, cut off their cases and mounted them with appropriate fuses and switches in a small aluminum minibox.

Everything worked fine until the repair warranty period expired, then the same problems started again.

I was about to break the inviolate Radio Shack seal, when I noticed that the expansion interface printed circuit (PC) board was badly warped. This was pulling the center of the RS-232-C connector down and causing the board to lose contact. *Voila!*

I tried a little "brogan maintenance." I pried the RS-232-C edge card connector with a screwdriver to get it seated against the board. This seemed to do the trick, and although the problem recurs every month or two, it reseats easily.

Software Problems

Hardware problems were not my only obstacles in getting the Spinwriter to run. I also had numerous software problems, especially when I tried to step up beyond 300 baud.

I was about to write a software driver patch for both the

Reader Service—see page 226

Electric Pencil and the KVP software driver I had purchased, when I decided to take a look at the Radio Shack RS-232-C driver software.

When I read the description of the driver routine in Radio Shack's manual, it mentioned that the software tests the DATA SET READY (DSR) for a low and loops each time if not. This is definitely not true, as you can see if you examine the listing of the driver on pages 27-28. The driver does test the UART status. It must in order to avoid losing characters.

While trying to find a way to patch all three of my printer driver routines – KVP, Electric Pencil and Radio Shack's – I realized that I could test both the UART and the printer status line at the same time. In other words, if either line is active, then the driver routine does not transmit.

After disassembling the driver routines for all three software drivers, I found that each routine tests the UART status in exactly the same way.

Success at 1200 Baud

At this point, I abandoned my attempts to patch each piece of software and started to concentrate on modifying my RS-232-C board to tie a printer status line with the UART status line. The modifications I made are shown in Figs. 1 and 2.

I connected the printer status line (Spinwriter pin 19) to the DSR line (pin 6) from the RS-232-C board. The printer status line is inverted and changed from RS-232-C levels to TTL levels by U3. The Spinwriter printer Reverse Channel line (line 19) was used as the printer status line. This line can be set to go either high or low when any of the following occur:

1. Buffer 7/8 Full
2. Paper Out
3. Ribbon End
4. Check Condition
5. Cover Open
6. Parity/Framing Error

With the Reverse Channel line set to go high when any of the above occur, pin 11 of U3 can be tied directly to the UART status line (pin 22 of U6 or pin 4 of U5). Although a piece of wire can be used for this connection, I used a small switching diode, a 1N914, which cost all of 11 cents. The anode should be connected to U3 pin 11. I used a small piece of wire wrap to connect the cathode to U5 pin 4.

With this simple modification, the printer driver routines test data line 6 for a logic 1, and if either the UART is busy or the printer status line is high, the printer driver stays in a loop. Essentially, both lines are ORed together.

I now run my NEC Spinwriter at 1200 baud. The print speed is more than double the 300 baud speed, because the internal logic of the NEC can process multiple spaces extremely fast. When multiple spaces are encountered, the print head moves directly to the next printable character, as soon as it is received from the serial line. Overall throughput is dramatically increased. ■

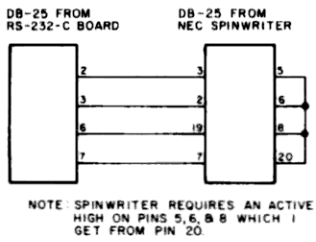


Figure 1.

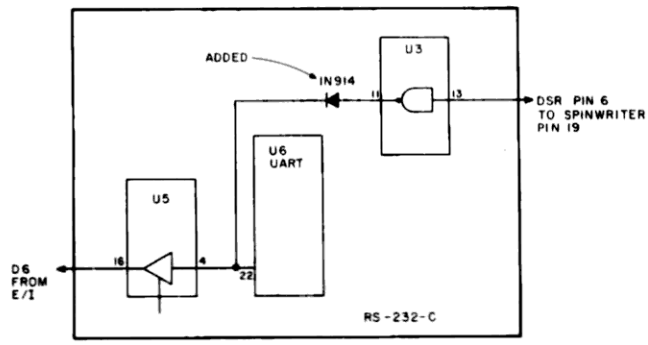


Figure 2.

✓ Reader Service—see page 226

A pioneering simulation of the trek to Fort Stinkendosert.

Westward Ho!

Raymond J. Herold
8363 Shady Grove Circle
Manassas, VA 22110

Westward, Ho! simulates the westward trek by the earliest, and possibly the bravest, of the pioneers. The program does not reflect any particular journey or trail, but rather a composite of all the varied dangers and situations the pioneers faced.

To give the program a competitive flavor, up to four pioneers can play at one time. They engage in a race to see who can reach their destination first, assuming anyone survives! A single player attempts to make the journey in the fewest number of days. The journey is not easy, and many decisions you make early on affect your success.

When the game begins, you are asked to enter the number of players and the name of each.

Choosing your Supplies

Each player picks the supplies he wants to take on the journey. You face the same dilemma as the pioneers: What do you take into the unknown? You want to be prepared for any situation, yet each additional item adds weight to the wagon. The more weight the horses must pull, the less distance you go each day.

The longer you are in the wilderness, the greater the odds of meeting a tragic end. Since you only have the opportunity to select supplies once, choose carefully! Keep in mind that the total weight you carry decreases as you consume food and water. Each player starts with a team of four horses.

At the beginning of each player's turn, a map is drawn detailing the area he must cover to reach his destination. Different graphics characters are used to symbolize deserts, forests and mountains. The remaining areas are prairies. Your starting point (in the East) and destination (in the West) are shown, along with an indicator showing your current location.

When the map is drawn, approximately 6-7 seconds elapse before locations are printed. The computer uses this time to figure out where you are, so be patient.

Use the map as an aid in selecting the route you wish to follow. There are advantages and disadvantages to any direction you may choose. You use more water in the desert, but it is a more direct route. If you go through the forests, food and water aren't as critical, but you must cover more ground. When you reach the mountains, you must find a pass in order to travel across them. If you are lucky, you may stumble across a water hole or stream, or even a lone settler who may help, but don't count on it! And, of course, there is danger everywhere.

After you have examined your map, you get a status report. This tells you where you are, how far you have traveled, how much water and food you have and your condition and that of the horses. Your condition and the condition of the horses falls into two categories.

First of all, if you or the horses are hungry, thirsty, wounded or sick, this will obviously affect how far you can travel. Secondly, if you are seriously sick or wounded, say, mauled by a bear, you certainly wouldn't be able to travel at all. In this instance you may lose several days of travel while you recuperate.

Scouting the Area

You have the option of scouting the area and finding out what is to the north, south, east

PP	Location <<print position
S$(s)	Name of each type of supply
G(s)	Initially, weight of each type of supply; Resumed as switches for condition of players
S(s,s)	Weight of each type of supply carried by each player
T(s)	Total weight carried by each player
M(s)	Contains codes for locations; dynamically generated for each players turn
X(s)	X coordinate for each player: value 1 to 40
Y(s)	Y coordinate for each player: value 1 to 24
N	Player number subscript
E,WY,EH,WH	Percent allocation for food and water
WT	Factor for overall food and water allocated
NF(s),NW(s),HW(s)	Keep track of number of days without food, water
SF(s)	Factor for players' overall condition
HH(s)	Factor for horses' overall condition
ML,MP,PL,FL,DL	Special location settings: mountains, Prairie, forest, desert
D(s)	Number of days of travel time lost
C	Miles traveled each day
MI(s)	Total miles traveled by each player
NH(s)	Number of horses for each player
D$	Direction traveled
L$(s)	Names of locations
M$(s)	Messages to player
C$	Condition names (sick, hungry, etc.)

In addition to the above, the following transient variables are used:
YY,WX,XW,YW,U,V,W,Z,I,TC,B,S0-S9,SW$,SS$,SN$,SE$,TW$,TS$,TN$,TE$

Table 1. Program variables.

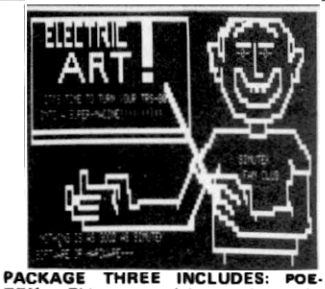
SOFTWARE → TRS-80 ← SOFTWARE

PACKAGE ONE INCLUDES: GRAPHIC-TREK "2000" — This full graphics, real time game is full of fast, exciting action! Exploding photon torpedoes and phasers fill the screen! You must actually navigate the enterprise to dock with the giant space stations as well as to avoid klingon torpedoes! Has shields, galactic memory readout, damage reports, long range sensors, etc! Has 3 levels for beginning, average, or expert players! ★ **INVASION WORG** — Time: 3099, Place: Earth's Solar System Mission: As general of Earth's forces, your job is to stop the Worg invasion and destroy their outposts on Mars, Venus, Saturn, Neptune, etc! Earth's Forces: Androids — Space Fighters — Lazer Cannon — Neutrino Blasters! Worg Forces: Robots — Saucers — Disintegrators — Proton Destroyers! Multi level game lets you advance to a more complicated game as you get better! ★ **STAR WARS** — Manuever your space fighter deep into the nucleus of the Death Star! Drop your bomb, then escape via the only exit. This graphics game is really fun! May the Force be with you! ★ **SPACE TARGET** — Shoot at enemy Ships with your missiles. If they eject in a parachute, capture them — or if you're cruel, destroy them! Full graphics, real time game! ★ **SAUCERS** — This fast action graphics game has a time limit! Can you be the commander to win the distinguished cross! Requires split second timing to win! Watch out!

ONLY 14.95

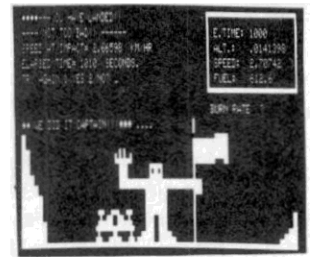
PACKAGE TWO INCLUDES: CHECKERS 2.1 — Finally! A checkers program that will challenge everyone! Expert as well as amateur! Uses 3-ply tree search to find best possible move. Picks randomly between equal moves to assure you of never having identical games. ★ **POKER FACE** — The computer uses psychology as well as logic to try and beat you at poker. Cards are displayed using TRS-80's full graphics. Computer raises, calls, and sometimes even folds! Great practice for your Saturday night poker match! (Plays 5 card draw). ★ **PSYCHIC** — Tell the computer a little about yourself and he'll predict things about you, you won't believe! A real mind bender! Great amusement for parties. ★ **TANGLE MANIA** — Try and force your opponent into an immobile position. But watch out, they're doing the same to you! This graphics game is for 2 people and has been used to end stupid arguments. (And occasionally starts them!) ★ **WORD SCRAMBLE** — This game is for two or more people. One person inputs a word to the computer while the others look away. The computer scrambles the word, then keeps track of wrong guesses.

ONLY 14.95



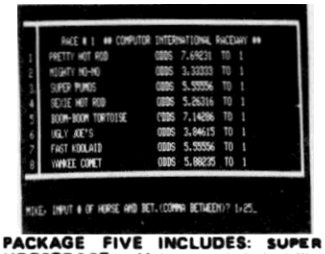
PACKAGE THREE INCLUDES: POETRY — This program lets you choose the subject as well as the mood of the poem you want. You give TRS-80 certain nouns or names, then the mood, and it does the rest! It has a 1000-word + vocabulary of nouns, verbs, adjectives and adverbs! ★ **ELECTRIC ARTIST** — Manual: draw, erase, move as well as, Auto: draw, erase and move. Uses graphics bits not bytes. Saves drawing on tape or disk! ★ **GALACTIC BATTLE** — The Swineus enemy have long range phasers but cannot travel at warp speed! You can, but only have short range phasers! Can you blitzkrieg the enemy without getting destroyed! Full graphics — real time! ★ **WORD MANIA** — Can you guess the computer's words using your human intuitive and logical abilities? You'll need to, to beat the computer! ★ **AIR COMMAND** — Battle the Kamikaze pilots. Requires split second timing. This is a FAST action arcade game.

ONLY 14.95



PACKAGE FOUR INCLUDES: LIFE — This Z-80 machine language program uses full graphics! Over 100 generations per minute make it truly animated! You make your starting pattern, the computer does the rest! Program can be stopped and changes made! Watch it grow! ★ **SPACE LANDER** — This full graphics simulator lets you pick what planet, asteroid or moon you wish to land on! Has 3 skill levels that make it fun for everyone. ★ **GREED II** — Multi-level game is fun and challenging! Beat the computer at this dice game using your knowledge of odds and luck! Computer keeps track of his winnings and yours. Quick fast action. This game is not easy! ★ **THE PHARAOH** — Rule the ancient city of Alexandria! Buy or sell land. Keep your people from revolting! Stop the rampaging rats. Requires a true political personality to become good! ★ **ROBOT HUNTER** — A group of renegade robots have escaped and are spotted in an old ghost town on Mars! Your job as "Robot Hunter" is to destroy the pirate machines before they kill any more settlers! Exciting! Challenging! Full graphics!

ONLY 14.95



PACKAGE FIVE INCLUDES: SUPER HORSERACE — Make your bets just like at the real racetrack! 8 horses race in this spectacular graphic display! Up to 9 people can play! Uses real odds but has that element of chance you see in real life! Keeps track of everyone's winnings and losses. This is one of the few computer simulations that can actually get a room of people cheering! ★ **MAZE MOUSE** — The mouse with a mind! The computer generates random mazes of whatever size you specify, then searches for a way out! The second time, he'll always go fastest route! A true display of artificial intelligence! Full graphics, mazes & mouses! ★ **AMOEBA KILLER** — You command a one man submarine that has been shrunken to the size of bacteria in this exciting graphic adventure! Injected into the president's bloodstream, your mission is to destroy the deadly amoeba infection ravaging his body! ★ **LOGIC** — This popular game is based on Mastermind but utilizes tactics that make it more exciting and challenging — has 2 levels of play to make it fun for everyone. ★ **SUBMARINER** — Shoot torpedoes at the enemy ships to get points. Fast action graphics, arcade type game is exciting and fun for everybody!

ONLY 14.95

HARDWARE → TRS-80 ← HARDWARE

MICRO SPEED

Upgrade your "slow" TRS-80 to a **SUPER FAST MACHINE!! (2.66 MHZ) over 50% FASTER!** Some of the features:

Auto turn-off during cassette or disk access. (This means NO lost programs EVER!) (Turns back on automatically too!) MANUAL control. (Unit may be turned on or off at any time. Yes even during program execution!) Keyboard indicator light "blinks" when micro-speed is on. Stops blinking when off! **Don't wait for SARGON II or any other program!!!** Comes with easy to follow instructions. (Some soldering required.) **OR** take to your local computer store or TV-Appliance Center for quick installation. (5-10 minutes!!) Works with any model, TRS-80.

ONLY 29.95 complete

MICRO BEEP

Simple hook up: Just plug cassette remote jack into unit.

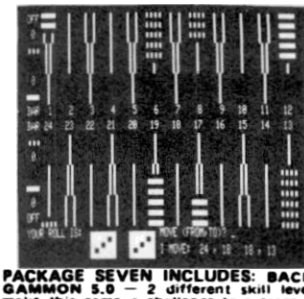
EASILY CONTROLLED FROM BASIC:

OUT 255,4 = on
OUT 255,0 = off

MICRO-BEEP make games more fun as well as provide useful sound output for professional applications!

Works with Any Model I TRS-80

ONLY 15.95 complete



PACKAGE SEVEN INCLUDES: BACKGAMMON 5.0 — 2 different skill levels make this game a challenge to average or advanced players FAST (15 second avg) Looks for best possible move to beat you! FANTASTIC GRAPHICS. Plays doubles and uses international rules ★ **SPEED READING** — Increases your reading speed. Also checks for comprehension of material. Great for teenagers and adults to improve reading skills ★ **PT 109** — Drop depth charges on moving subs. Lower depths get higher points in this fast action graphics game. ★ **YAHTZEE** — Play Yahtzee with the computer. This popular game is even more fun and challenging against a TRS-80! ★ **WALL STREET** — Can you turn your $50,000 into a million dollars? That's the object of this great game Simulates an actual stock market!

ONLY 14.95

PACKAGE SIX INCLUDES: 20 HOME FINANCIAL PROGRAMS — Figures amortization, annuities, description rates, interest tables, earned interest on savings and much, much more. These programs will get used again and again. A must for the conscientious, inflation minded person.

ONLY 14.95

SIMUTEK

Reader Service—see page 226

and west. If you are lucky, you may even spot a stream or a house. You must find a pass to get through those mountains.

At this point, you must decide how much food and water you will consume, as well as how much water and oats you will provide for the horses. This is done by entering a number from 0 to 100, which represents a percent of the daily allocation. That is, 100 percent would be the amount you normally eat in one day; 50 percent would be half of that. While the wilderness is certainly no place to be gluttonous, the extent to which you ration affects your ability and the horses' ability to travel. Cutting the ration below certain levels for a prolonged period of time could result in tragedy.

You are also asked to select the direction in which you wish to travel. In the mountains, you may be unable to travel in a certain direction. Also, you would be well advised to heed "No Trespassing" signs.

Every now and then you encounter dangers or opportunities along the way. In some instances, you have a choice in what happens. In other cases, your fate is sealed by how wisely you chose your supplies before the journey began.

Program Execution

Table 1 shows the variables and their functions. Variables with an (s) refer to a subscripted item.

When the program begins execution, the names of the supplies are loaded into the S$(s) variables. The respective weights are loaded into G(s). As each player picks his supplies, the weights are stored in the appropriate S(s,s) location. After all players have picked their supplies, the G(s) variables are cleared, to be used later as indicators for player and horses conditions. The ML, MP, PL, FL, and DL variables are loaded at the start of the program with special location addresses. These are actual video memory addresses and point to such items as water holes, streams, and mountain passes.

Miles traveled by each player each day is determined by multiplying a percentage of the total weight carried by factors for water and food consumed by you and the horses, your aggregate condition and the horses' aggregate condition and the number of horses. The following equation is used:

$$C = ((((143-T(N))/4 + 10)*WT)*NH(N)*((SF(N) + HH(N))/2)$$

The most critical aspect of the program is the ability to keep track of each player's location on the map. To accomplish this, a 40 x 24 virtual matrix is used. The term virtual is used to point out the fact that no actual memory is allocated to the matrix. Only the actual coordinates of each player's location are stored in the X and Y variables.

To see how this works, visualize a 40 x 24 matrix. Each element in the matrix represents 10 miles. A player's current location might be X = 30, Y = 20. If he travels west 30 miles, then the formula X = X – (30/10) gives his new location of 27,20. Traveling east, the distance would be added to X; south the distance would be added to Y; north the distance would be subtracted from Y. The area taken by the map is 60 characters across and 12 lines down. Using the following formula correlates our imaginary matrix coordinates to the 1024 print positions on the video monitor.

```
PP = INT((int((Y(N)/2) + .5)*64)
     + (X(N)*1.5) + 64)
     PRINT@PP,"<<"
```

We now know where to print the << indicator. But how do we know what lies in each direction from the current location? By adding the value of PP to the beginning of video memory (15360), we have the player's location as an actual video memory address. Usng Z as the current video memory location we PEEK(Z) for 146 (desert), 170 (forest) or 188 (mountains). Depending on the location, a one-digit code (0,1,2,3) is loaded into M(0), which is the variable for the current locaiton. We do the same for the six matrix locations (60 miles) in each direction in the following manner:

```
West    M(1) thru M(6)  -
    Z-1,Z-2,Z-3,Z-4,Z-5,Z-6
East    M(7) thru M(12) -
Z+1,Z+2,Z+3,Z+4,Z+5,Z+6
North   M(13) thru M(18) -
Z-64,Z-128,Z-192,Z-256,Z-320,Z-384
South   M(19) thru M(24) -
Z+64,Z+128,Z+192,Z+256,Z+320,Z+384
```

The values of M(s) can then be examined to determine what is in any given direction. ■

Program Listing.

```
1 ' WESTWARD, HO!
2 ' COPYRIGHT (C) 1979   RAY HEROLD
3 '
50 CLEAR130:DIM S$(20),G(20):GOSUB6000:FORX=1TO20:READS
     $(X):NEXTX:FORX=1TO20:READG(X):NEXTX
100 RANDOM:DIM S(20,4),M(24)
150 CLS:PRINT@64,"NUMBER OF PLAYERS 1 - 4";:INPUTP:IFP<
     1ORP>4THEN150
200 FORX=1TOP:PRINT:PRINT"NAME OF PLAYER NUMBER";X;:INP
     UTN$(X):X(X)=40:Y(X)=14:NH(X)=1:NEXTX
300 FORN=1TOP:GOSUB1000:NEXTN
400 FORX=1TO20:G(X)=0:NEXTX
500 FORN=1TOP:S1=0:IFSF(N)=9THENNEXTN:GOTO500:ELSEGOSUB
     5000:GOSUB2000:GOSUB3000:NEXTN:ND=ND+1:GOTO500
1000 CLS:PRINTTAB(15)N$(N);" - SELECT YOUR SUPPLIES":PR
     INTTAB(10)"NUMBER AFTER ITEM INDICATES WEIGHT":PRI
     NT:FORX=18TO20:G(X)=S(X,N):NEXTX
1010 FORX=1TO10:PRINTX;"- ";S$(X);G(X);:PRINTTAB(30)X+1
     0;"- ";S$(X+10);G(X+10):NEXTX
1020 PRINT@896,"ENTER ITEM YOU WISH TO TAKE.   0 IF FIN
     ISHED";:INPUTI:IFI<0ORI>20THEN1020
1030 IFI=0RETURN
1035 IFS(I,N)<>0PRINT@896,"YOU ALREADY HAVE ";S$(I);STR
     ING$(45," ");:FORX=1TO800:NEXTX:GOTO1020
1040 IFI<18S(I,N)=G(I):T(N)=T(N)+G(I):GOTO1000
1050 L=63-(LEN(S$(I))+23):PRINT@896,"HOW MANY DAYS WORT
     H OF ";S$(I);STRING$(L," ");:PRINT"(5 TO 25 DAYS)";
     :D=99:INPUTD:IFD>4ANDD<26S(I,N)=D:T(N)=T(N)+D:GOTO
     1000:ELSEGOTO1050
2000 ' STATUS
2005 GOSUB2200:GOSUB2400:GOSUB2600
2006 FORZ=18TO20:IFS(Z,N)<0S(Z,N)=0:NEXTZ:ELSENEXTZ
2007 CLS:PRINT:PRINTTAB(15)N$(N);"'S CURRENT STATUS":PR
     INT
2010 PRINT"YOUR LOCATION:   ";L$:PRINT"YOU HAVE TRAVELE
```

Program continues

```
     D";MI(N);"MILES IN";ND;"DAYS.":PRINT"YOUR FOOD SUP
     PLY IS";S(18,N);"DAYS.":PRINT"YOU HAVE";S(19,N);"D
     AYS OF OATS FOR THE HORSES."
2020 PRINT"YOU HAVE A WATER SUPPLY OF";S(20,N);"DAYS.":
     PRINT"YOU ARE CARRYING";T(N);"POUNDS OF SUPPLIES."
     :PRINT"YOU ARE:     ";CD$(1);CD$(2);CD$(3);CD$(4)
2030 PRINT"THE HORSES ARE:    ";HD$(1);HD$(2):PRINTSTRIN
     G$(64,"-"):IFS9=1RETURN
2050 PRINT@896,"SCOUT THE AREA - Y OR N","";:INPUTA$:I
     FA$="Y"GOSUB2700
2060 GOSUB2300:W1=E/100:W2=(WY*.2)/100:W3=(WH*.8)/100:W
     4=EH/100:WT=W1+W2+W3+W4:TH=WT:WT=WT/3:IFD(N)>0THEN
     2099
2070 PRINT"WHICH DIRECTION WILL YOU GO  -  N, S, E, W";
     :INPUTD$:IFD$<>"N"ANDD$<>"S"ANDD$<>"W"ANDD$<>"E"TH
     EN2070
2074 C=((((143-T(N))/4)+10)*WT)*NH(N)*((SF(N)+HH(N))/2)
2079 IF((D$="N"ANDS3=1)OR(D$="S"ANDS4=1)OR(D$="E"ANDS5=
     1)OR(D$="W"ANDS6=1))ANDC>10GOSUB2100:GOTO2099
2080 IFD$="S"GOSUB2900
2081 IFD$="N"GOSUB2920
2082 IFD$="W"GOSUB2940
2083 IFD$="E"GOSUB2960
2085 MI(N)=MI(N)+C
2099 T(N)=T(N)-TH:S(18,N)=S(18,N)-W1:S(19,N)=S(19,N)-W4
     :S(20,N)=S(20,N)-(W3+W2):RETURN
2100 ' LOCATION STOP
2110 IFD$="S"Y(N)=Y(N)+2
2120 IFD$="N"Y(N)=Y(N)-2
2130 IFD$="W"X(N)=X(N)-.666
2140 IFD$="E"X(N)=X(N)+.666
2150 MI(N)=MI(N)+10:RETURN
2200 ' LOCATION
2205 IFX(N)<-1ORX(N)>41ORY(N)<0ORY(N)>26CLS:PRINT@64,"T
     HERE'S A SIGNPOST UP AHEAD:
YOU HAVE JUST ENTERED
     THE TWILIGHT ZONE!":X(N)=40:Y(N)=14:INPUT"PRESS EN
     TER";A$:GOSUB5000:RETURN
2210 IFX(N)<1ORX(N)>40ORY(N)<1ORY(N)>24THENL$="YOU ARE
     LOST!!":RETURN
2215 IFM(0)=23L$=M$(5):RETURN
2220 IFM(0)>4L$=M$(M(0)-10)ELSEL$=L$(M(0))
2299 RETURN
2300 ' SUPPLIES
2310 CLS:PRINT@64,"IT'S TIME TO DIVVY UP THE GRUB, PARD
     NER!":PRINT:PRINT"HOW MUCH OF THE DAILY RATION WIL
     L YOU CONSUME  (0 - 100%)":PRINT:Z=((N-1)*5)+1
2315 IFS(18,N)>0INPUT"FOOD FOR YOU";E:NF(N)=NF(N)-1:ELS
     EINPUT"YOU ARE OUT OF FOOD!!!";A$:E=0
2320 IFE<0ORE>100THEN2315
2325 IFS(20,N)>0PRINT:INPUT"WATER FOR YOU";WY:HW(N)=HW(
     N)-1:NW(N)=NW(N)-1:ELSEPRINT:INPUT"YOU ARE OUT OF
     WATER!!!";A$:WY=0:WH=0:GOTO2340
2330 IFWY<0ORWY>100THEN2325
2335 PRINT:INPUT"WATER FOR THE HORSES";WH:IFWH<0ORWH>10
     0THEN2335
2340 IFS(19,N)>0PRINT:INPUT"OATS FOR THE HORSES";EH:HF(
     N)=HF(N)-1:ELSEPRINT:INPUT"YOU ARE OUT OF OATS!!!"
     ;A$:EH=0
2345 IFEH<0OREH>100THEN2340
2346 IFE<40NF(N)=NF(N)+1:G(Z)=1
2347 IFWY<30NW(N)=NW(N)+1:G(Z+1)=1
2348 IFEH<40HF(N)=HF(N)+1:HC(U)=1
2349 IFWH<30HW(N)=HW(N)+1:HC(U+1)=1
2350 IFLS<0ORLS>762THEN2360
2354 IFM(0)=1ORM(0)=11WY=WY*1.3:WH=WH*1.5:EH=EH*1.2
2356 IFM(0)=3ORM(0)=13EH=EH*.6
2358 IFM(0)=2ORM(0)=12EH=EH*.4
2360 IFNF(N)<0NF(N)=0ELSEE=E*(1+(NF(N)/10))
2362 IFNW(N)<0NW(N)=0ELSEWY=WY*(1+(NW(N)/10))
2364 IFHW(N)<0HW(N)=0ELSEWH=WH*(1+(HW(N)/10))
2366 IFHF(N)<0HF(N)=0ELSEEH=EH*(1+(HF(N)/10))
2399 CLS:RETURN
2400 ' CONDITION
2410 FORX=1TO4:CD$(X)="":NEXTX:Z=((N-1)*5)+1:TT=0:TC=0
2430 FORX=ZTOZ+3:TC=TC+G(X):NEXTX:IFTC=0CD$(1)="O.K.":G
     OTO2499
2440 FORX=ZTOZ+3:B=(X-Z)+1:IFG(X)>0G(X)=G(X)-1:CD$(B)=C
     $(B):TT=TT+1:NEXTX:ELSENEXTX
2499 SF(N)=1-(TT/4):RETURN
2500 W=INT((Z/2)+.5):RETURN
2510 W=INT(J*1.5):RETURN
2550 CLS:PRINT@64,"YOU MUST FIND ";M$(3);" TO TRAVEL
IN
     THAT DIRECTION":PRINT:INPUT"PRESS ENTER";A$:RETUR
     N
2600 ' HORSE'S COND.
2610 HD$(1)="":HD$(2)="":U=((N-1)*2)+1:TT=0
2620 IFHC(U)+HC(U+1)=0HD$(1)="O.K.":GOTO2640
2630 FORX=UTOU+1:B=(X-U)+1:IFHC(X)>0HC(X)=HC(X)-1:HD$(B
     )=C$(B):TT=TT+1:NEXTX:ELSENEXTX
2640 HH(N)=1-(TT/4):RETURN
2700 ' SCOUT AREA
2703 CLS:TW$="":TS$="":TN$="":TE$="":SW$="":SS$="":SN$=
     "":SE$="":IFS(16,N)=0THENPRINT:PRINT"YOU AREN'T GO
     NNA DO MUCH SCOUTING AROUND WITHOUT
FIELD GLASSES,
     TINHORN.  NOT TOO SMART!!":INPUTA$:RETURN
2705 S3=0:S4=0:S5=0:S6=0
```

Program continues

✓Reader Service—see page 226

```
2706 IFM(13)=23A=3:SS$=M$(5):GOTO2715
2707 IFM(13)>4ANDM(13)<>14SS$=M$(M(13)-10):S4=1
2710 IFY(N)>23TS$=WM$:GOTO2716
2711 IFM(14)=23A=3:GOTO2715
2712 IFM(14)>4A=M(14)-10ELSEA=M(14)
2715 TS$=L$(A)
2716 IFM(1)=23A=3:SW$=M$(5):GOTO2725
2718 IFM(1)>4ANDM(1)<>14SW$=M$(M(1)-10):S6=1
2720 IFX(N)<2TW$=WM$:GOTO2726
2721 IFM(2)=23A=3:GOTO2725
2722 IFM(2)>4A=M(2)-10ELSEA=M(2)
2725 TW$=L$(A)
2726 IFM(19)=23A=3:SN$=M$(5):GOTO2735
2728 IFM(19)>4ANDM(19)<>14SN$=M$(M(19)-10):S3=1
2730 IFY(N)<2TN$=WM$:GOTO2736
2731 IFM(20)=23A=3:GOTO2735
2732 IFM(20)>4A=M(20)-10ELSEA=M(20)
2735 TN$=L$(A)
2736 IFM(7)=23A=3:SE$=M$(5):GOTO2745
2738 IFM(7)>4ANDM(7)<>14SE$=M$(M(7)-10):S5=1
2740 IFX(N)>39TE$=WM$:GOTO2780
2741 IFM(8)=23A=3:GOTO2745
2742 IFM(8)>4A=M(8)-10ELSEA=M(8)
2745 TE$=L$(A)
2780 PRINT:PRINT"TO THE NORTH:  ";TN$:PRINT"  ";SN$:PRI
     NT"TO THE EAST:  ";TE$:PRINT"  ";SE$:PRINT"TO THE
     SOUTH:  ";TS$:PRINT"  ";SS$:PRINT"TO THE WEST:  ";
     TW$:PRINT"  ";SW$:INPUT"ENTER";A$:RETURN
2900 Z=Y(N):YW=Z:YY=0:GOSUB2500:WA=W:Z=(Z+(C/10)):K=Z:G
     OSUB2500:WB=W
2902 FORZ=WA+1TOWB:W=(Z-WA)+12
2910   IFM(W)=23GOSUB2550:GOSUB2980:Y(N)=Y(N)+YY:RETURN
     :ELSEYY=YY+2
2915 GOSUB4000:NEXTZ:Y(N)=K
2916 IFYW<24ANDK>24Y(N)=24
2919 RETURN
2920 Z=Y(N):YW=Z:YY=0:GOSUB2500:WA=W:Z=(Z-(C/10)):K=Z:G
     OSUB2500:WB=W
2922 FORZ=WA-1TOWBSTEP-1:W=(WA-Z)+18
2930   IFM(W)=23GOSUB2550:GOSUB2980:Y(N)=Y(N)-YY:RETURN
     :ELSEYY=YY+2:
2935 GOSUB4000:NEXTZ:Y(N)=K
2936 IFYW>1ANDK<1Y(N)=1
2939 RETURN
2940 J=X(N):XW=J:WX=0:GOSUB2510:WA=W:J=(J-(C/10)):K=J:G
     OSUB2510:WB=W
2942 FORJ=WA-1TOWBSTEP-1:W=WA-J
2950   IFM(W)=23GOSUB2550:GOSUB2990:X(N)=X(N)-WX:RETURN
     :ELSEWX=WX+.666
2955 GOSUB4000:NEXTJ:X(N)=K
2956 IFXW>1ANDK<1X(N)=1
2959 RETURN
2960 J=X(N):XW=J:WX=0:GOSUB2510:WA=W:J=(J+(C/10)):K=J:G
     OSUB2510:WB=W
2962 FORJ=WA+1TOWB:W=(J-WA)+6
2970   IFM(W)=23GOSUB2550:GOSUB2990:X(N)=X(N)+WX:RETURN
     :ELSEWX=WX+.666
2975 GOSUB4000:NEXTJ:X(N)=K
2976 IFXW<40ANDK>40X(N)=40
2979 RETURN
2980 C=YY*10:RETURN
2990 C=WX*15:RETURN
2999 RETURN
3000 ' SITUATIONS
3005 CLS:PRINT
3007 IFNF(N)>5ORNW(N)>5ORHF(N)>5ORHW(N)>5PRINTN$(N);" D
     IED ON THE JOURNEY.
R.I.P.":PRINT:INPUT"ENTER";A$:
     PC=PC+1:SF(N)=9:IFPC=PTHENPRINT"THE GAME IS OVER."
     :STOPELSERETURN
3010 IFNF(N)>3PRINT"YOU ARE STARVING":S0=1
3012 IFNW(N)>3PRINT"YOU ARE DYING OF THIRST":S0=1
3014 IFHW(N)>3PRINT"THE HORSES ARE DYING OF THIRST":S0=
     1
3016 IFHF(N)>3PRINT"THE HORSES ARE STARVING":S0=1
3020 IFS0=1INPUT"ENTER";A$:S0=0:CLS
3021 IFD(N)>0THEND(N)=D(N)-1:RETURN
3022 IFM(0)=11ORM(0)=12PRINT:PRINT"THERE IS WATER HERE.

DO YOU WANT TO FILL THE CONTAINERS?  Y OR N";:INP
     UTF$:IFF$="Y"T(N)=T(N)+(25-S(20,N)):S(20,N)=25
3025 IFM(0)=10ANDS(15,N)>0PRINT:PRINT"THE SETTLER'S HER
     E WILL GIVE YOU 5 DAYS WORTH
OF OATS FOR YOUR GOLD
      DUST.  DO YOU ACCEPT  Y - N";:INPUTF$:IFF$="Y"S(1
     9,N)=S(19,N)+5:S(15,N)=0:T(N)=T(N)+5
3030 IFT(N)<65ANDND=2PRINT"ONE OF YOUR HORSES STUMBLED
     AND BROKE HIS LEG.
YOU HAD TO DESTROY HIM.":PRINT:
     INPUT"ENTER";A$:NH(N)=NH(N)-.3:RETURN
3040 IFRND(15)=7ANDM(0)<4ANDM(0)<>1PRINTR$(M(0)):PRINTQ
     $:ELSEGOTO3100
3045 A$="":INPUTA$:IFA$="N"RETURN
3046 IFS(9,N)<1INPUT"YOU DON'T HAVE ANY AMMO.
ENTER";A$
     :RETURN
3050 V=0:IFS(7,N)<>0V=8ELSEIFS(8,N)<>0V=14
3055 IFV=0INPUT"YOU DIDN'T BRING A RIFLE OR GUN.
ENTER"
```

Program continues

```
     ;A$:RETURN
3060 IFRND(V)>6INPUT"YOU MISSED HIM!
ENTER";A$:RETURN:E
     LSEINPUT"YOU GOT HIM!!
ENTER";A$:S(18,N)=S(18,N)+4
     :T(N)=T(N)+4:RETURN
3100 IF(T(N)<80)V=3ELSEV=5
3110 IFRND(V)<>3RETURN
3120 IFM(0)>3THEN3199
3130 V=(4*M(0))+RND(4):ONVGOSUB3200,3300,3350,3500,3270
     ,3300,3370,3400,3280,3300,3350,3450,3200,3250,3280
     ,3400
3199 RETURN
3200 PRINT"IT GETS COLD OUT HERE AT NIGHT.":IFS(3,N)>0A
     NDS(4,N)>0PRINTM$(7);S$(3);" AND ";S$(4):INPUT"ENT
     ER";A$:RETURN
3210 IFS(3,N)<1PRINTM$(6);S$(3)
3212 IFS(4,N)<1PRINTM$(6);S$(4)
3215 PRINT"YOU CAUGHT A BAD COLD":INPUT"ENTER";A$:G(((N
     -1)*5)+3)=3:RETURN
3250 PRINT"IT'S VERY ROCKY HERE.":IFS(1,N)>0PRINTM$(7);
     S$(1):INPUT"ENTER";A$:RETURN
3260 PRINTM$(6);S$(1):PRINT"YOU TWISTED YOUR ANKLE":INP
     UT"ENTER";A$:G(((N-1)*5)+4)=3:RETURN
3270 PRINT"YOUR WAGON HIT A GOPHER HOLE":IFS(17,N)>0PRI
     NTM$(7);S$(17):PRINT"IT TAKES 1 DAY TO REPAIR":D(N
     )=1:INPUT"ENTER";A$:RETURN:ELSEPRINTM$(6);S$(17):P
     RINT"IT TAKES 3 DAYS TO REPAIR":D(N)=3:INPUT"ENTER
     ";A$:RETURN
3280 PRINT"YOU ARE APPROACHED BY A  BEAR":IFS(6,N)>0PRI
     NT"HE TAKES YOUR ";S$(6);" AND LEAVES":S(6,N)=0:T(
     N)=T(N)-3:INPUT"ENTER";A$:RETURN
3285 IFRND(2)=1ANDS(18,N)>6PRINT"HE TAKES HALF YOUR FOO
     D AND LEAVES":V=S(18,N)/2:T(N)=T(N)-V:S(18,N)=S(18
     ,N)-V:INPUT"ENTER";A$:RETURN:ELSEPRINT"HE MAULS YO
     U.  YOU TAKE 3 DAYS TO RECOVER":D(N)=3:G(((N-1)*5)
     +4)=5
3286 INPUT"ENTER";A$:RETURN
3300 IFM(0)=1V=1ELSEV=2
3310 PRINT"WATCH OUT!
YOU ARE BITTEN BY ";B$(V):IFS(10,
     N)>0PRINTM$(7);S$(10):PRINT"YOU TAKE 2 DAYS TO REC
     OVER":G(((N-1)*5)+3)=3:D(N)=2:INPUT"ENTER";A$:RETU
     RN
3315 PRINTM$(6);S$(10):IFS(11,N)>0PRINT"THE WHISKEY HEL
     PS SOME.
YOU TAKE 3 DAYS TO RECOVER.":G(((N-1)*5)+
     3)=4:D(N)=3:ELSEPRINT"YOU TAKE 4 DAYS TO RECOVER":
     D(N)=4
3316 G(((N-1)*5)+3)=5:INPUT"ENTER";A$:RETURN
3350 PRINT"INDIANS!!!":IFS(14,N)>0PRINT"YOU OFFER THEM
     ";S$(14):IFRND(4)<>2PRINT"THEY ACCEPT":INPUT;A$:S(
     14,N)=0:T(N)=T(N)-4:INPUTA$:RETURN:ELSEPRINT"THEY
     DONT LIKE TRINKETS.
THEY TAKE ONE OF THE HORSES":N
     H(N)=NH(N)-.2:INPUTA$:RETURN
3360 IFS(9,N)>0PRINT"THEY TAKE YOUR AMMO":S(9,N)=0:T(N)
     =T(N)-3:INPUTA$:RETURN:ELSEPRINT"YOU FIGHT THEM OF
     F BUT ARE WOUNDED":G(((N-1)*5)+4)=4:INPUTA$:RETURN
3370 PRINT"YOU ARE ATTACKED BY A COYOTE.":IFS(5,N)>0PRI
     NT"YOU USE YOUR KNIFE TO FIGHT HIM OFF.
YOU ARE SL
     IGHTLY WOUNDED":G(((N-1)*5)+4)=2:INPUTA$:RETURN
3375 PRINT"YOU MAKE IT BACK TO THE WAGON BUT YOU'RE
SER
     IOUSLY WOUNDED.
YOU TAKE 3 DAYS TO RECOVER":D(N)=3
     :G(((N-1)*5)+4)=6:INPUTA$:RETURN
3400 PRINT"YOU ARE BESET BY BANDITS":IFS(15,N)>0PRINT"T
     HEY TAKE YOUR GOLD DUST":S(15,N)=0:T(N)=T(N)-5:INP
     UTA$:RETURN
3410 IFS(11,N)>0PRINT"THEY TAKE YOUR WHISKEY":S(11,N)=0
     :T(N)=T(N)-5:INPUTA$:ELSEPRINT"YOU FIGHT THEM OFF
     BUT ARE WOUNDED":G(((N-1)*5)+4)=3:INPUTA$:RETURN
3420 IFRND(3)=2ANDS(20,N)>6PRINT"THEY GET DRUNK AND SHO
     OT UP THE WATER TANKS.
YOU LOSE HALF OF IT.":V=S(2
     0,N)/2:T(N)=T(N)-V:S(20,N)=S(20,N)-V:INPUTA$
3425 RETURN
3450 PRINT"YOUR WAGON GETS STUCK IN THE MUD":IFS(12,N)>
     0PRINTM$(7);S$(12):PRINT"YOU LOSE 1 DAY":D(N)=1:IN
     PUTA$:RETURN:ELSEPRINTM$(6);S$(12)
3455 IFS(13,N)>0PRINT"YOU HAVE TO CUT DOWN A TREE WITH
     YOUR AXE
TO MAKE LEVERS.
IT TAKES 2 DAYS TO GET OU
     T.":D(N)=2:ELSEPRINT"IT TAKES 4 DAYS TO GET OUT":D
     (N)=4
3460 INPUTA$:RETURN
3500 PRINT"A DUST STORM HITS":IFS(2,N)>0PRINTM$(7);S$(2
     ):PRINT"YOU STOP FOR 1 DAY TO LET IT PASS":D(N)=1:
     INPUTA$:RETURN:ELSEPRINT"THE SAND INJURES YOUR EYE
     S
YOU TAKE 3 DAYS TO RECOVER YOUR SIGHT":G(((N-1)*
     5)+4)=4:D(N)=3:INPUTA$:RETURN
3999 RETURN
4000 IFM(W)=14S9=1:L$=M$(4):GOSUB2007:PRINTTAB(20)"-- T
     HE WINNER --":STOP:ELSERETURN
5000 ' DRAW MAP
```

Program continues

✓Reader Service—see page 226

```
5010 IFS2=0GOSUB5800:S2=1
5020 CLS:PRINTTAB(25)"MAP FOR ";N$(N):PRINTSTRING$(63,C
     HR$(143));:FORX=127TO896STEP64:PRINT@X,CHR$(191);:
     NEXTX
5030 FORX=64TO832STEP64:PRINT@X,CHR$(191);:NEXTX:PRINT:
     PRINTSTRING$(64,CHR$(143));
5040 Y=0:Z=0:B=6:FORX=494TO886STEP64:Y=Y+1:IFY=3THENZ=1
5050   IFY=4ORY=5Z=5:B=11
5055   IFY=6ORY=7Z=6:B=8
5057   IFY>7Y=4
5060   PRINT@X-Z,STRING$(B,CHR$(146));
5070 NEXTX:FORX=648TO840STEP64:Z=0:B=8:PRINT@X,STRING$(
     8,CHR$(146));:NEXTX
5080 Y=0:Z=0:B=7:FORX=170TO618STEP64:Y=Y+1:IFY=2THENZ=2
5090   IFY=3Z=3:B=10
5095   IFY=4Z=5:B=12
5097   IFY=5Z=6:B=6
5098   IFY=6Z=5:B=4
5100   PRINT@X-Z,STRING$(B,CHR$(170));
5105 NEXTX:Z=5:FORX=194TO514STEP64:PRINT@X,STRING$(Z,CH
     R$(170));:Z=Z+1:NEXTX:FORX=578TO660STEP64:B=3:Z=0:
     PRINT@X,STRING$(3,CHR$(170));:NEXTX
5110 Y=0:Z=0:B=7:FORX=145TO860STEP64:Y=Y+1
5115   IFY>3Z=Z+1:IFY>8THENB=13ELSEB=9
5120   PRINT@X+Z,STRING$(B,CHR$(188));
5130 NEXTX:
5160 PRINT@567,"START *";:PRINT@769,"* END";
5200 PP=INT((INT((Y(N)/2)+.5)*64)+(X(N)*1.5)+64):GOSUB5
     500
5210 PRINT@747,"DESERT";:PRINT@296,"FOREST";:PRINT@665,
     "MOUNTAINS";:PRINT@985,"PRESS ANY KEY";:IFPP>0PRIN
     T@PP,"<<";
5299 IFINKEY$=""THEN5299ELSERETURN
5500 FORW=0TO24:M(W)=0:NEXTW
5510 FORW=0TO6:Z=(15360+PP)-W
5512   IFPEEK(Z)=146M(W)=1
5514   IFPEEK(Z)=170M(W)=2
5516   IFPEEK(Z)=188M(W)=3
5518 WZ=0:GOSUB5700:NEXTW
5520 FORW=1TO6:Z=(15360+PP)+W
5522   IFPEEK(Z)=146M(W+6)=1
5524   IFPEEK(Z)=170M(W+6)=2
5526   IFPEEK(Z)=188M(W+6)=3
5528 WZ=6:GOSUB5700:NEXTW
5530 FORW=1TO6:Z=(15360+PP)+(W*64)
5532   IFPEEK(Z)=146M(W+12)=1
5534   IFPEEK(Z)=170M(W+12)=2
5536   IFPEEK(Z)=188M(W+12)=3
5538 WZ=12:GOSUB5700:NEXTW
5540 FORW=1TO6:Z=(15360+PP)-(W*64)
5542   IFPEEK(Z)=146M(W+18)=1
5544   IFPEEK(Z)=170M(W+18)=2
5546   IFPEEK(Z)=188M(W+18)=3
5548 WZ=18:GOSUB5700:NEXTW:RETURN
5700 FORWW=0TO10
5710   IFZ=MP(WW)M(W+WZ)=13
5715   IFZ=ML(WW)M(W+WZ)=23
5720 NEXTWW
5725 FORWW=1TO4
5730   IFZ=PL(WW)M(W+WZ)=10
5735   IFZ=DL(WW)M(W+WZ)=11
5740   IFZ=FL(WW)M(W+WZ)=12
5742   IFZ=STTHENM(W+WZ)=14
5745 NEXTWW:RETURN
5800 ML(1)=15509:ML(2)=15573:ML(3)=15637:ML(4)=15703:ML
     (5)=15768:ML(6)=15833:ML(7)=15898:ML(8)=15963:ML(9
     )=16032:ML(10)=16097:ML(0)=16162:MP(1)=16227:MP(2)
     =15701:MP(3)=15702:MP(4)=16028:MP(5)=16029:MP(6)=1
     6030:MP(7)=16031
5805 MP(8)=15638+(RND(5)*65):ST=16129
5810 PL(1)=15565+(RND(4)*65):PL(2)=15566+(RND(4)*65):PL
     (3)=15580+(RND(4)*65):PL(4)=15583+(RND(4)*65)
5820 DL(1)=15728+(RND(4)*64):DL(2)=15917+(RND(4)*64):DL
     (3)=15944+(RND(4)*65):DL(4)=15945+(RND(4)*64)
5830 FL(1)=15466+(RND(5)*65):FL(2)=15466+(RND(5)*63):FL
     (3)=15559+(RND(5)*65):FL(4)=15560+(RND(5)*65)
5890 RETURN
6000 ' INITIALIZE
6060 L$(0)="PRAIRIE":L$(1)="DESERT":L$(2)="FOREST":L$(3
     )="MOUNTAINS":M$(0)="A LITTLE HOUSE ON THE PRAIRIE
     ":M$(1)="A WATER HOLE":M$(2)="A STREAM":M$(3)="A M
     OUNTAIN PASS"
6065 M$(5)="AN IMPASSABLE CLIFF":C$(1)="HUNGRY,":C$(2)=
     "THIRSTY,":C$(3)="SICK,":C$(4)="WOUNDED,"
6070 WM$="A SIGN. IT SAYS:
** NO TRESSPASSING! **":M$(4
     )="A SMALL TOWN":L$(4)=M$(4)
6080 R$(0)="THERE'S A PRAIRIE DOG UP AHEAD.":R$(2)="THE
     RE'S A SQUIRRELL UP IN A TREE.":R$(3)="THERE'S A R
     ABBIT IN THE BUSHES.":Q$="DO YOU WANT TO SHOOT IT
     FOR FOOD  Y - N?":M$(6)="YOU SHOULD HAVE BROUGHT "
     :M$(7)="GOOD THING YOU BROUGHT "
6090 B$(1)=" A SCORPION":B$(2)=" A RATTLESNAKE"
6999 RETURN
7000 DATA"BOOTS","A BANDANA","BLANKETS","LONG JOHNS","A
      KNIFE","ROCK CANDY","A RIFLE","A REVOLVER","AMMUN
     ITION","MEDICAL SUPPLIES","WHISKEY","ROPE","AN AXE
     ","A BOX OF TRINKETS","GOLD DUST","FIELD GLASSES",
     "SPARE WAGON PARTS"
7050 DATA"FOOD FOR YOU","OATS FOR THE HORSES","WATER",4
     ,1,3,1,3,3,7,5,3,5,5,2,4,4,5,4,15,0,0,0
```

INFLATION FIGHTER—ROUND # 2

KO Inflation With Our Knock-Out Prices

MPI 88T Impact Matrix Printer

Quality, Full-Page Printout For Your TRS 80 Computer!

SAVE $150!

$599

Reg. List $749

Unrivaled champion of the small business, educational, personal computing and professional user. Quality construction and continuous duty print head allow heavy usage. Attractive styling complements the most elegant of systems without sacrificing compact size.

● Type of Printing: **Impact bidirectional 7x7 dot matrix** ● Print Rate: **100 characters per second** (maximum) ● Thruput: 80 characters per second (maximum) ● Character Set: Full **upper and lower case 96 character ASCII set,** software selectable single or double wide character fonts ● Character Height: 0.10 in. (0.25 cm) ● Print Format: 8.0 in. (20.3 cm) line length, **80** characters per line at 10 CPI, **96** characters per line at 12 CPI, **120** characters per line at 15 CPI, **132** characters per line at 16.5 CPI ● Paper Feed: 10 lines per second, stepper motor controlled. User selectable pressure roller or tractor feed ● Line Spacing: **6 or 8 lines per inch,** user selectable ● Media: **Roll paper:** 8.5 in. (21.6 cm) wide by 5 in. (12.7 cm) diameter single ply or pressure sensitive multiple copy paper, 0.012 in. (3 mm) maximum thickness. **Fan Fold paper:** 1 in. (10.1 cm) to 9.5 in. (24.1 cm) sprocket (including sprocket margins), 0.012 inc. (3 mm) maximum thickness. **Cut Sheet paper:** Maximum width, 9.5 in. (24.1 cm) ● Ribbon: Continuous loop cartridge, 20 yds. 0.5 in. (1.27 cm) wide black ribbon, 5 million character line ● Input Power: 115/230 VAC. ± 10%, 50/60 HZ ● Data Input: **Parallel:** Centronics compatible 7-bit ASCII, TTL levels with strobe, acknowledge returned to indicate data was received. **Serial:** RS232C or 20 ma Current Loop with BUSY (RS232C only) handshake, 10 or 11 bits; 100, 150, 300, 600, 1200 baud ● **Data Buffer: 1K (2K optional)** ● **Forms Control:** Top of Form (eight selectable forms lengths) Skip over perforation ● Physical Dimensions: 16.25 in. (41.3 cm) wide x 10.75 in. (27.3 cm) deep x 6.25 in. (15.9 cm) high. Dimensions exclude paper and paper holder. Weight: less than 15 lbs. (6.75 Kg)

ACCESSORIES

I/O Cable (Specify Computer Type and Serial or Parallel)	$19
Extra Ribbon Cartridge	$9
2K Buffer	$39
Roll Paper Holder	$24

Other Contenders . . .

PRINTERS

NEC 5510 RO w/tractor List $2950	$2595
NEC 5520 KSR w/tractor List $3270	$2950
Diablo 1650 RO w/tractor List $3425	$2890
Diablo 1650 KSR w/tractor List $3895	$3285
TI 810 Basic Serial List $1895	$1645
TI 810 Basic Parallel List $1940	$1695
TI 743 KSR u/c ASCII List $1395	$895
Anadex DP9500 200 cps List $1650	$1449
Centronics 702 RO 120 cps List $2440	$1995
Centronics 703 RO 120 cps List $3140	$2395
Centronics 704 RO 180 cps List $2350	$1885
Sanders Media 12/7 50-200 cps List $4100	$3265

$$ SUPER VALUES $$
(Equivalent or better performance than Radio Shack TRS80 Line Printer III)

ANACOM 150 List $1395 **NOW $1195**

* 150 cps, bidirectional Logic Seeking * 80, 132 or 136 columns * 6 or 8 lines per inch * 5.5 ips slew speed * 9x9 Matrix upper and lower case with decenders * 10 char/inch * 5" to 14 7/8" fan fold paper, tractor feed * original plus 5 copies * 6 million character life snap-in ribbon cartridge * 120/240 VAC 50/60HZ power * Size 23" x 14" x 8" (58.4 cm x 35.6 cm x 20.3 cm metric) weight 30 lbs. (38 lbs. shipping)

DATA ROYAL 5000

80 Column List $1295 ..... $1150
136 Column List $1395 ..... $1250

* 96 ASCII Characters * 125 cps * 6 lines per inch * 9x9 Matrix * Selectable expanded characters * True upper and lower case plus underlining * Short line capability * Dynamic Platen with adjustable character density * Prints original plus 5 copies * Front or bottom paper feed * Tractor feed * 5 IPS paper slew (independent of head motion) * Top of form * Ribbon cartridge * Parallell or Serial (110 to 9600 Baud) * Quietized cabinet

COMPUTERS

NEW! From Personal MicroComputers Inc.

PMC-80 Computer $579
Funtionally equivalent to TRS80 Model I Computer with Level II Basic and 16K Memory. Has built in cassette recorder, TV modulator, TV monitor output plus additional features

PMC-80 Fastload Cassette System $289
Loads standard Level II Basic or System Cassette tapes 16 times faster at 8000 Baud. Includes modified cassette and I/O adapter

TRS-80 ADD-ONS

FCI-80 FastLoad Cassette Interface $149
Unlike other high speed cassette decks, our FCI loads in standard format cassettes at 16 times the normal speed. Yes, you can load "Blackjack" in less than 4 seconds. Plugs into the back of your keyboard or expansion interface and operates on ROM based program under TRS-80 control, called from Basic, System or Disk Basic. The built-in ROM also provides keyboard debounce, auto repeat and key-beep. A modified CTR-41 cassette tape recorder is used allowing play and fast forward buttons to latch down during read of tape. It can still be used for CSAVE at normal speed. Comes with instruction booklet for modification of CTR-41. Powerpack and TRS-80 interconnect cable extra.

REX-80 ROM EXTENDER $39
Enables use of the 2014 empty address locations between the end of Basic ROM and the TRS-80 RAM. It fully decodes them so there is no interference when a 2048 Byte ROM is used. Allows interchangeable ROMs with commonly used routines and programs to be accessed by System or USR command. Internal jumpers allow use of most industry standard 24-pin plus 5 volts ROMs. A connector cable attaches REX-80 to 40-pin bus connector at back of keyboard or expansion box. All signals routed through REX-80 are buffered (except data) to allow expansion for other devices off its output connector. Power pack and TRS-80 interconnect cable extra.

CAB-40 Flat Cable/40 PIN $25
PWR-80 Power Pack (9V at 500 ma.) $8

NEW

MT 32—Printer/Memory Expansion Module. Drive any centronics compatible printer and add 16K or 32K of RAM.

Basic Unit $99
With 16K $159
With 32K $199

NEW

MPC Microconnection List $300 ..... $275
An intergrated RS232 adapter and direct connect modem for any model TRS-80. Connects directly to computer bus eliminating a need for Radio Shack Expansion Box, RS232 Adapter and Acoustic interface. Complete with bus connector and intelligent terminal software program.

CYBERNETICS TRS-80 — CP/M BUSINESS SOFTWARE

The fastest Mod-II CP/M with the most features!!!

MOD-II CP/M $249
* Over 610,000 bytes/disk * Downloading package included * 1,000 baud operation of serial printer without data loss * Single drive backup.

MOD-I CP/M $149
* Mixed single/double density on any of 4 drives (even a 1-drive system) * Ultra fast disk operation * Emulation of cursor addressing for any of several "dumb" CRTs.

CBASIC-2 (Mod I or II) $109
* Auto L-F printer support & ASCII top of form software. **(LPHIII)** * Supplemental document describing our implementation * User-settable function keys

The following software for Mod-II CP/M only unless otherwise stated (requires CBASIC-2):

RM/COBOL Only COBOL for CP/M with alternate keys (multikey ISAM), CRT screen handling, interactive debug, Z80 code, and the most useful Level 2 features **Compatible with Tandy's Cobol—but runs faster!** $495

PMS (Property Management System) Interactive, menu-driven system includes full G/L, budgeting, cash journal, delinquency list, tenant activity/rent roll, complete audit trail and reports on vacancies, lost rent, and vendors $650
Demo disk and manual ..... $75.

APH (Automated Patient History) General purpose question-asking, answer-printing system furnished as self-administered review of systems general patient history (Mod-I also) $175

Magic Wand Full feature word processing, true proportional spacing, file merging, and use of full-screen editor for source programs or data $399

RPA (Residential Property Analysis) Analyzes income and expense, financing, taxes, inflation and depreciation on home, condo, or apartments over a user-selectable time. Shows payoff in terms of ROI, Cap rate, cash-on-cash. Amortization schedules and worksheet $299

RBC (Rent/Buy Comparison) Sales or investment tool to compare renting and savings account investment vs. purchasing a particular property $249

Osborne CBASIC source programs (Mod-I also):

Payroll w/cost accounting $249
Accts. Payable/Accts. Receivable $249
General Ledger w/cash journal $249
O&A (CBASIC Books (ea.) ..... $20.

GAMES

Adventure $14
Bridge Challenger $14
Star Trek III $14
Cribbage $10
Sargon Chess $19
Sargon II (Cassette) $29
(Disk) $33
Hunt $19

TERMS: Cash, check or money order, bank wire transfer. C.O.D. or credit cards. $10.00 minimum. Charge orders must include expiration date. Purchase orders also accepted from recognized institutions. Include telephone number with all orders. Advertised prices are for prepaid orders. F.O.B. shipping point. Charge and credit orders add 2%. CODs required 25% deposit. California residents add 6½% sales tax. For shipping in U.S. add ($2.50 min.) 2% West of U.S., 3% East of Mississippi, otherwise freight collect (air service where applicable). Foreign orders must be accompanied by payment in U.S. funds and include 10% for shipping. Quantities may be limited. Retail prices vary from mail order. All prices subject to change and all offers subject to withdrawal without notice. All equipment is new with manufacturers warranty unless otherwise indicated.

PALOMAR COMPUTER PRODUCTS ✓420

Call or Write For Free Catalog
(714) 744-7314/744-9595

910 W. San Marcos Blvd. # 105, San Marcos, California 92069

An exciting laboratory application for the people's computer.

DVM Interface for the 80

Karl J. Casper
Harry R. Freedman
Department of Physics
Cleveland State University
Cleveland, OH 44115

For recording scientific measurements, the TRS-80 is little better than a hand-held calculator if all of the data is entered through the keyboard. The 80 becomes a scientific instrument only when data can be read from a port. But in some cases, this data moves so swiftly that it cannot be read by a BASIC program.

An example of this is the information recorded by a digital voltmeter. The settling time — the time for the voltmeter to stabilize at a particular reading — is usually in the range of 0.3 to 1.0 seconds, which can conceivably be read in BASIC. But the digits in digital voltmeters are often strobed at rates of 500 Hz or greater — the appearance of a continuous display is only a consequence of the persistence of human vision.

No BASIC program can read information into the computer at this rate. Rather, a machine language program is required, preferably one that can be easily linked to a BASIC program. When this linking is possible, the experimenter can write flexible programs in a high level language to analyze and display the data recorded through the machine language program.

In the past few years, single-chip peripheral interface adapters have been developed, such as the 8255, which we have used to interface a Keithley Model 179 digital voltmeter to the TRS-80. The circuit attached to the TRS-80 requires only two additional chips and a third chip is mounted inside the Keithley DVM. All the connections are made by cable to the edge card connector on the rear of the 80's keyboard.

This same circuit can be used to interface any other relay-driven devices to the TRS-80 with suitable programming instructions. You can use it to monitor an air conditioner or in a security system to shut off and turn on lights, automatically.

The Interface

Digital voltmeters not only come in all shapes and sizes; the digital information is displayed with different numbers of digits and by different decoding schemes. In liquid crystal displays, for example, all segments of a seven-segment display are continuously available from the conversion circuit. A total of 35 to 40 lines, depending on the decimal point display, may be physically connected to the display circuit. Decoding this information with a TRS-80 requires a multiplex circuit that reads each digit in succession, a straightforward task, but with more wiring than we wanted.

The Keithley Model 179 digital voltmeter, like many voltmeters using light emitting diodes, strobes each digit into the display circuit. The lighted output is not continuous: Each digit is illuminated in turn for two milliseconds, then turned off for eight milliseconds. Each digit is illuminated in turn while the other digits are off.

Multiplexing a display in this way not only saves power, but the digital information is ideal for reading directly into the TRS-80. The time period of two milliseconds is more than long enough to read the data and store it in a memory location. Moreover, the information is in BCD form; the conversion to ASCII is trivial.

Photo 1 shows the front panel of the Keithley DVM. The voltage is measured in ranges from 0.2 volts to 2000 volts, the current in ranges from 0.2 mA to 2 amps, and the resistance from 2000 ohms to 20 megohms. The sign is automatically displayed.

This model DVM has no external connector to the TRS-80. We installed a 25-pin connector on the back of the DVM and used cables between it and the interface box. Inside we added a board with one integrated circuit to wire from the range switches to the connector. Although this is simple, it does void the Keithly warranty.

Basically, the 8255 peripheral interface adapter (PIA) has three ports — any of which may be used as inputs and outputs — and three modes of operation. In the first mode, called mode 0, each of the three eight-bit ports, A, B and C, can be programmed for either input or output by writing the appropriate control word to the PIA. There is no handshaking and any port can be switched between input and output functions by writing a different control word to the PIA.

Mode 1 permits input/output data to be transferred to or from a specific port with handshaking. For example, a device may strobe the PIA informing it that the device has data to be read. The microcomputer, which is programmed to poll Port C looking for a strobe, reads the data into the input buffer of either

Port A or Port B. The PIA can then output the data to another external device in a similar fashion.

Mode 2 permits bi-directional communication with a peripheral device. Port C is used to determine the direction of data flow in both Ports A and B.

Initially, we connected the interface for mode 0 operation. However, this required a separate flip-flop which latched the TRS-80 into a Wait state until the data strobes arrived. There was nothing wrong with this as long as the data strobes were physically present to bring the TRS-80 out of this Wait condition.

We worried that the memory might not be refreshed, but Radio Shack kindly informed us that the Z-80 memory refresh cycle continued even in this state. The Z-80 simply executes NOPs during Wait to activate the refresh register.

However, of the eight different TRS-80's we connected to this interface, one unit apparently did not refresh during Wait, but did perform all other functions correctly.

Since it is unlikely that most users will have checked the Wait function, and since the additional flip-flop can be eliminated in mode 1, we abandoned mode 0 operation.

Though the techniques we use do not take full advantage of handshaking, the TRS-80 is committed to reading the DVM as quickly as possible once the subroutine is accessed and is not allowed to perform any other function. It constantly polls Port C for the strobe marking the next digit to be read. After reading and strobing this data (approximately 50 microseconds), the TRS-80 returns to polling Port C.

The 8255 Circuit

The complete interface circuit wiring diagram is shown in Fig. 1. The 8255, designated as IC2, is used in mode 1 operation.

Ports A and B are both inputs and Port C directs the interface. The PIA is addressed by the TRS-80 whenever the instruction OUT 240 through OUT 243 is used. This address is decoded by the eight input NAND gate, IC1, and the address lines A0 and A1.

Port A reads the digital data in BCD form. Since this information is desired in ASCII, the two high order bits PA4 and PA5 are connected directly to +5 volts. Thus, the number four is read as 34, which is decoded in ASCII as 4.

Port B reads all of the residual information about the sign of the number and the range. As we will see, it is important to know if the DVM is measuring resistance. Pin PB1 reads this information.

It is also important to know if the DVM is connected to the interface circuit. Pin PB0 measures the DC supply voltage to determine not only if it is connected, but also if it is turned on. If neither of these conditions is fulfilled, the program returns immediately to BASIC.

Of the three ports, Port C has the most work to do. Since the data from the DVM is stored in a string, it is necessary to deter-

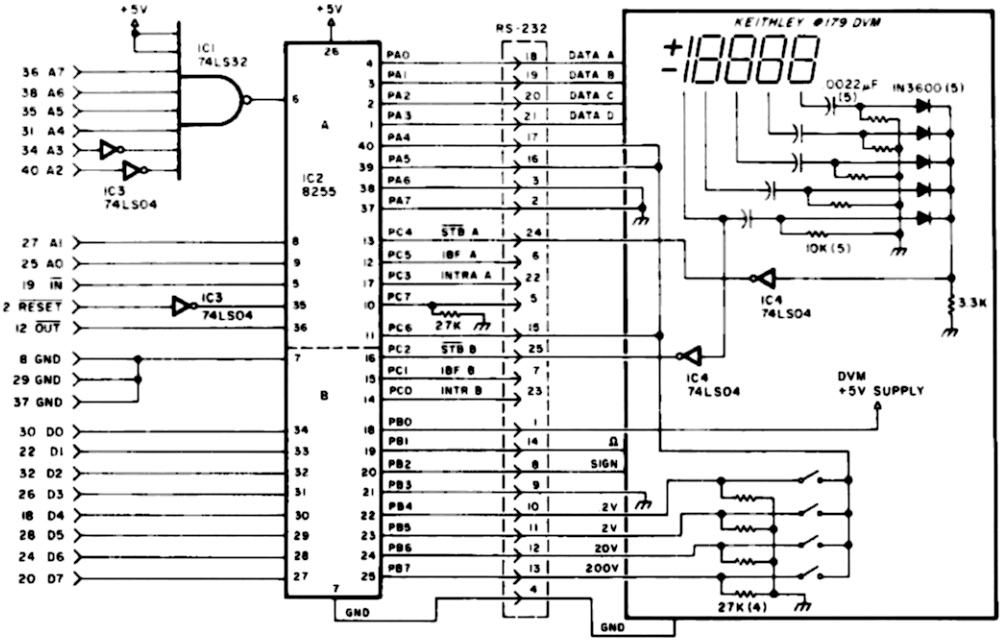
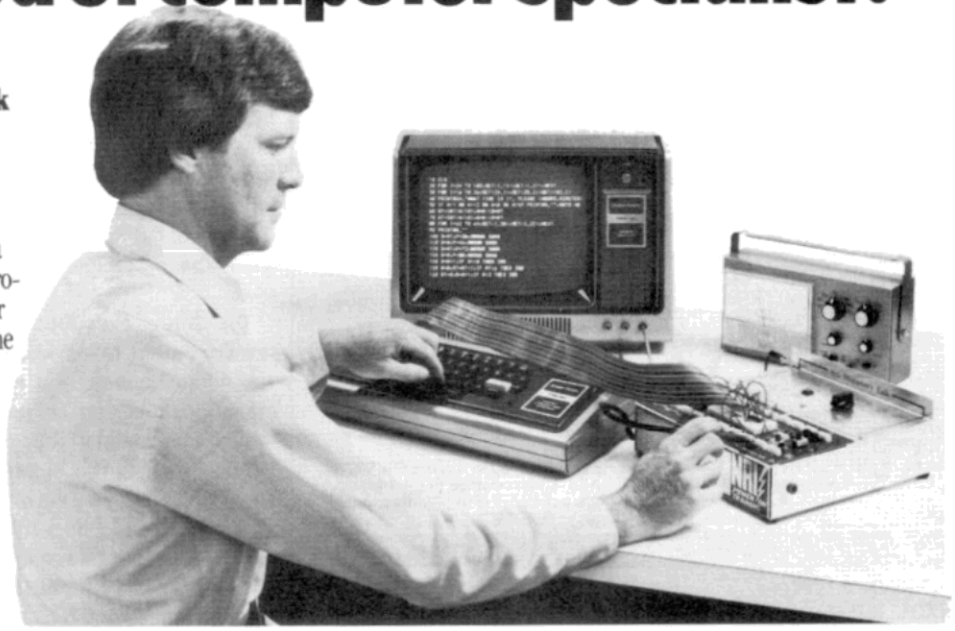


Fig. 1. Interface wiring diagram. The additional 5-volt power supply needed for the 8255 is not shown.

Now NRI takes you inside the world's most popular microcomputer to train you at home as the new breed of computer specialist!



NRI teams up with Radio Shack to teach you how to use, program and service microcomputers…make you the complete technician.

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 200,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and helping you over the tough spots.

Training includes TRS-80 computer, transistorized volt-ohm meter, digital frequency counter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

Explore the TRS-80 Inside and Out

NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you introduce and correct faults …watch how circuits interact…interface with other systems…gain a real insight into its nature.

You also build test instruments and the NRI Discovery Lab, performing over 60 separate experiments in the process. You learn how your trouble-shooting tools work, and gain greater understanding of the information they give you. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog… No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV and Audio, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
NO SALESMAN WILL CALL
Please check for one free catalog only.

All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council 179-100

mine the arrival of the first digit. The digit strobes are differentiated by RC circuits which are physically wired inside the DVM. Each circuit is an input for a diode OR circuit which then applies a signal to STB(B) at the arrival of a digit strobe—not necessarily the first one.

The first digit strobe is also connected to STA(A). Since this strobe is differentiated, this pin returns almost immediately to a logic 1 state, and, about 0.3 microseconds later, IBF(A) will be set to logic 1. If both IBF(A) and INTE(A) are high, then INTR(A) will be set to a logic 1 state. The computer can check the level of INTR(A) to determine when the first digit strobe arrives.

The actual execution of the main program takes less than 100 microseconds, and the remaining digits can be read in turn by checking INTR(B).

The PIA IC2, the eight-input NAND gate IC1, and the inverter IC3 are mounted in a minibox with a separate power supply as shown in Photo 2. About 60 mA total current is required by these chips, and neither the TRS-80 nor the Keithley DVM have that much to spare. A separate +5 volt power supply was used.

The interface box is attached by ribbon cables to the rear of the TRS-80 and to the Keithley DVM to a 40 pin jack that we mounted on the rear of the voltmeter. In this way, the adapter can be used for interfacing other equipment to the TRS-80.

When operating the three PIA ports as input, it is advisable to return the pins to ground through a resistor as is shown for pins PB4-PB7. We chose the 27000 ohm resistors because we had so many of them in our stock, but somewhat smaller or larger resistors would work just as well.

The decimal is not read as input data by Port A. The information is obtained through each range switch input on Port B. Moreover, we assume that the user knows whether volts, amps, or ohms are being measured. The wiring and the program are simplified by that assumption.

If the voltmeter is set to ohms, then PB1 will be set high, but that particular piece of information is determined only to place the decimal point correctly. The sign is determined by the logical value at pin PB2. If it is positive, then that point must register a logical 1.

Finally, pin PB0 determines that the digital voltmeter is turned on by checking its power supply. If the power is off, the program returns to BASIC with a reading of 0. This pin is continually polled during the execution of the program. If the five-volt power supply is off, no strobes appear and the program finds itself in an endless loop.

Using a PIA simplifies the circuit enormously. Only four integrated circuits are actually needed, and the software control subroutine is concise and easily written.

The PIA is controlled through Port 243. This is not real port, but selects the control word that defines the mode of the PIA ports (Fig. 2).

Selecting Ports A and B as input and both upper and lower halves of Port C as an output when the chip is operating in mode 1 means that the control word should be 10110110 or B6H. This mode continues until changed by writing a different control word to 243.

Imbedding the Subroutine

The machine language subroutine can be imbedded in the BASIC program by using string packing techniques. These were described among others by Mike Schmidt and Leo Christopherson in the May/June and July/Aug., 1979 issues of *80-U.S.* and by James Garon in the July, 1979 issue of the Orange Country Users Group Newsletter.

The machine code is literally stored as bytes in a character string in a BASIC statement. The starting address of this string is then passed on to the USR subroutine. Calling USR, accesses the machine language program.

The first step in the process is the creation of a program which will store the machine code as bytes in a string. The following statement defines a string vari-

able which must contain as many bytes as the machine language program:

```
9000 A$ = "**********1**********2****"
```

The line number 9000 is arbitrarily chosen, but to append this program to another, the line number should be larger than the other numbers in the BASIC program. The machine language subroutine is POKEd into the addresses corresponding to the bytes in the string. Each of the bytes represented by an asterisk or a number is replaced by one byte of the machine language program.

The input buffer, which interprets each line of a BASIC program before storing the line in memory, is only 255 bytes long. Therefore the machine language subroutine must not exceed 255 bytes, less the number of bytes needed for the line number and the characters A$ = " and the closing quotation mark.

The next step in the process picks up the beginning address of the string. The following statements are needed:

```
9010 A1 = PEEK (VARPTR(A$) + 1)
9020 A2 = PEEK(VARPTR(A$) + )
9030 A3 = A2*256 + A1
```

These statements set A1 equal to the least significant byte of the starting address of the string, A2 equal to the most significant byte of the address of the string, and A3 equal to the starting address of the string in decimal form.

The USR function in Level II BASIC looks for the least significant byte of the starting address of the machine language subroutine at location 16526 and the most significant byte of the address at 16527. The following statement passes the beginning address of the string to these locations:

```
9040 POKE 16526,A1 : POKE 16527,A2
```

Before storing these statements as a subroutine, the characters in the string must be replaced with the machine language program. After assem-

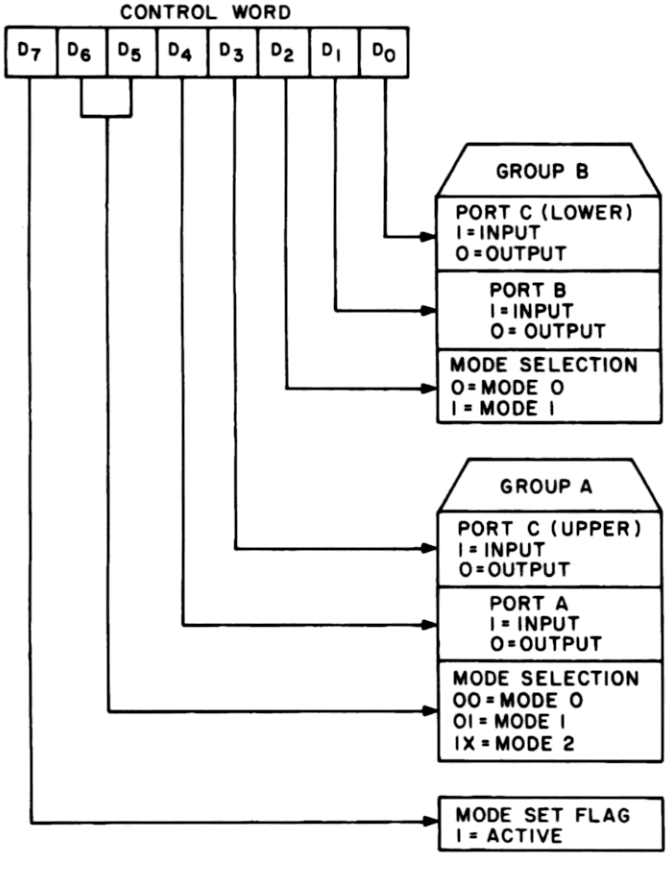


Fig. 2. Control word format for operation of 8255 peripheral interface adapter (Courtesy Intel Corp.).

bling the program using the Editor/Assembler, the machine language code is written in data statements in decimal form:

```
10010 DATA 217,221,229,245,175,40,1,1,1,1,
          1,1,1,1,1,1,79,6,5,221
10020 DATA 42,142,64,221,54,7,45,62,147,
          211,
          243,62,128,211,242,219,242,87,
          175,211
```

plus as many additional data statements as are necessary. Any number of bytes, up to approximately 250, can be entered into a single data statement, but editing a data statement containing more than 20 bytes is inconvenient and time consuming. In general, the actual number of bytes in a data statement and the format seems to vary widely according to individual preferences. The one used here is easy for us to edit and modify.

After writing N bytes into data statements, the following steps read them into the string:

```
10100 RESTORE
10110 FOR I = 0 TO N-1
10120 READ D : POKE A3 + I,D
10130 NEXT I
```

The complete program for reading the DVM subroutine into a string is shown in Program Listing 1. The first statement sets aside enough bytes for storing the string. This statement must exist in the BASIC program where this subroutine will be used.

When storing this subroutine on tape, only five steps, 9000 through 9040, will be retained. All other statements in the program in the listing may be deleted. The subroutine may form part of a general library of subroutines that are appended to a BASIC program. The symbols A1-A3 and A$ are arbitrary and may be changed and the line numbers may be renumbered at any time. Statement 9000 contains the machine language subroutine which is called by the statement X = USR(0). Any number of subroutines may be included in a program. It is only necessary to POKE the starting address of the subroutine into locations 16526 and 16527 each time that you wish to change the subroutine called by the USR function.

Even if the subroutine de-

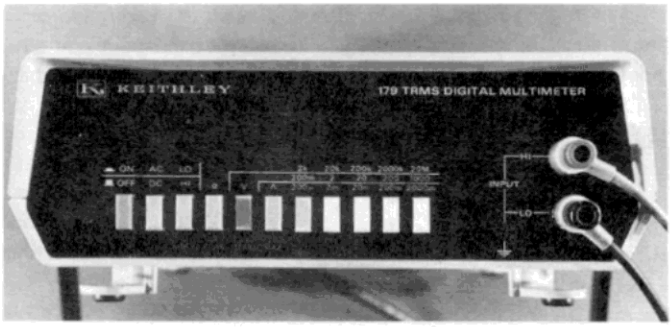


Photo 1. Front panel of Keithley digital voltmeter.

scribed here is the only one used in a program, it is desirable to access it in the following way:

```
2000 FOR N = 0 TO 99
2010 GOSUB 9000
2020 X = USR(0)
2030 NEXT N
```

The important step is line 2010. If any BASIC statement is inserted into the program at any time, the BASIC interpreter automatically reallocates memory storage of the rest of the BASIC program including the string containing the subroutine. The validity of the addresses that have been so carefully POKEd into locations 16526 and 16527 is lost. To restore that validity, the beginning address must be rePOKEd into those memory locations. Although it adds some time to the program, lines 2010 and 2020 should always be coupled in writing BASIC programs.

Subroutines from Disk

If the program is to be used with Disk BASIC, then, as Garon has pointed out, it is necessary to replace line 9040 with 9040 DEFUSRO = A3.

The subroutine can then be called with the statement X = USRO(0). Subroutines stored in Disk BASIC are even more easily appended to BASIC programs. The greater storage capacity and speed of the disk simplify programming, and line renumbering is a utility usually available in most disk operating systems. The procedure is quite similar to that used in large computers when accessing library subroutines.

After the data statements are read and the code POKEd into the string, line 9000 now has bytes in the string that BASIC interprets in several ways. For example, suppose that line 9000 originally set aside 10 bytes for a subroutine consisting of the following: 49,50,51,52,53,54,55, 56,57,58. This subroutine does nothing, but after POKEing the bytes into the string, line 9000 would look like this: 9000 A$ = "123456789:". These are the ASCII equivalents of the bytes in the string.

The Level II BASIC interpreter checks each byte in a BASIC statement to determine if it is:

A) A control code;
B) An ASCII character;
C) A token;

The second edition of the Level II BASIC reference manual provides more information about this procedure on pages C/2 and 3, and page E/1. When the interpreter sees a byte between 21H and 5FH, it interprets that byte as an ASCII character in the way shown in this illustration.

One byte in this group must be avoided, 22H. This is interpreted as the ASCII character and signifies the end of the string.

For bytes between 80H and FAH, and for the byte FFH, the interpreter sees that bit 7 is set and understands that the byte is a token for one of the BASIC functions. This results in less memory being needed in Level II for storing a BASIC program and gives an interesting appearance to the string.

Unfortunately, the second edition of the Level II manual overlooks the token for 80H (128D) which is END and the token for 255 which is ISA. If the data statement in the example contains the code 229,213,195,

219,241,211,243,217,209,201, then the resulting string would appear as:

```
9000 A$ = PEEK = ERRINPCDBLORLENA
     BS(KEY$"
```

all of which are tokens for this particular machine code.

Real problems begin when the interpreter encounters the bytes FB, FC, FD and particularly FE. All of these have a destructive appearance when the program is listed although the machine code remains intact with FB, FC and FD. For most scientific programs it is better to work around opcodes using these bytes, thereby making it simpler to debug and modify the programs. For these bytes and the following, the effects can be observed by POKEing the following bytes into line 9000:

```
10000 DATA 49,58,51,52,CC,53,CC,54,CC,
      55
```

For CC, simply substitute the desired bytes, and after POKEing, list line 9000.

Similar problems occur when the interpreter encounters bytes below 20H. From the Level II manual, these bytes are used for control codes for I/O functions. Trouble starts with byte 0. The BASIC interpreter, seeing this byte, thinks that it has reached the end of a line and that the next byte is the beginning of a new line. Machine code containing this byte must be avoided at all times.

The next byte 01 causes no problems, since the BASIC interpreter simply ignores it in displaying the string, although the machine code is stored in the correct memory locations. For example, if the data statement contains the bytes:

```
10000 DATA 49,50,51,52,1,53,1,54,1,55
```

then after reading this into the string line 9000 will have the appearance: 9000 A$ = "1234567"

The 01 byte has been ignored and it appears that the string only has seven characters. But when the memory locations corresponding to the string are examined, all of the bytes are found in the right places.

The use of bytes seen as control codes can be summarized as in Table 1. All bytes are shown in decimal form.

Machine Language Subroutine

In constructing the machine language routine with the TRS-80 Editor/Assembler, we followed several principles. First, the subroutine is to be imbedded in the BASIC program which dynamically reallocates it to different portions of memory and obviates setting aside any specific portion of memory.

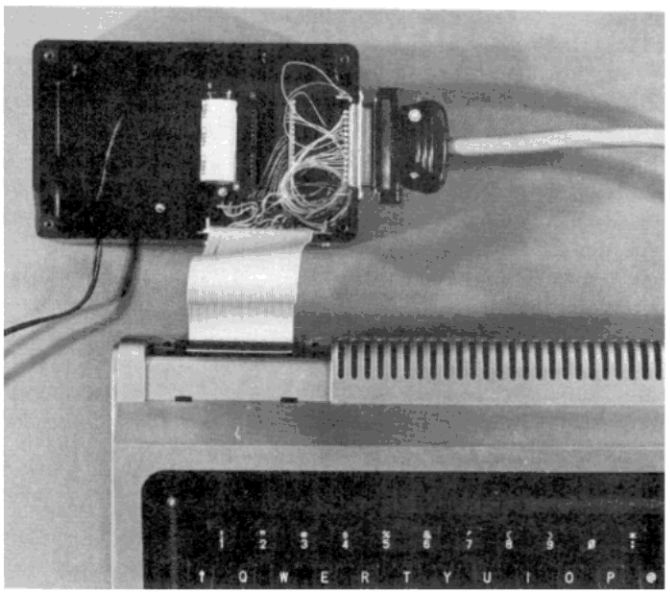


Photo 2. Minibox containing complete interface circuit. Box is larger than necessary for future interfaces.

ΩMEGA SALES CO.

"WHOLESALE COMPUTER PRICES" DIRECT TO THE PUBLIC

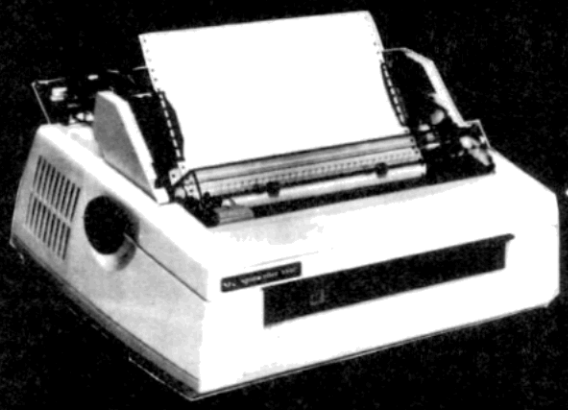
12 Meeting St., Cumberland, R.I. 02864

TRS·80
Model II - $3,500

Apple II
16K - $999

INTERTEC SUPERBRAIN
32K RAM - $2449.00
64K RAM - $2649.00

PRODUCT SPECIAL OF THE MONTH!!



NEC 5510 Printer
$2395
Tractor Feed Option
$180

Products are NOW IN STOCK AT ΩMEGA Sales Co.

Epson TX-80 - $699

Atari 800
$749

Soroc 120 - $699

CALL TOLL FREE FOR ΩMEGA'S PRICE!

ΩMEGA OFFERS THE BEST DELIVERY AND PRICE ON:
APPLE • ATARI • TRS·80 MODEL II • INTERTEC •
T.I. 810 • HEWLETT-PACKARD-85 • SOROC •
COMMODORE • NEC • QUME • CENTRONICS

ΩMEGA sells only factory fresh, top quality merchandise to our customers.
ΩMEGA will try to match any current advertised price with similar purchase conditions.
Before you buy anywhere else - be sure to call ΩMEGA Sales Co.
1-401-722-1027 or

389 **ΩMEGA TOLL FREE: 1-800-556-7586**

Overseas Inquiries Invited TELEX: 952106
ΩMEGA ships via UPS, truck, or air. COD's, VISA, Mastercharge accepted.

★★★ A PERCOM BULLETIN ★★★

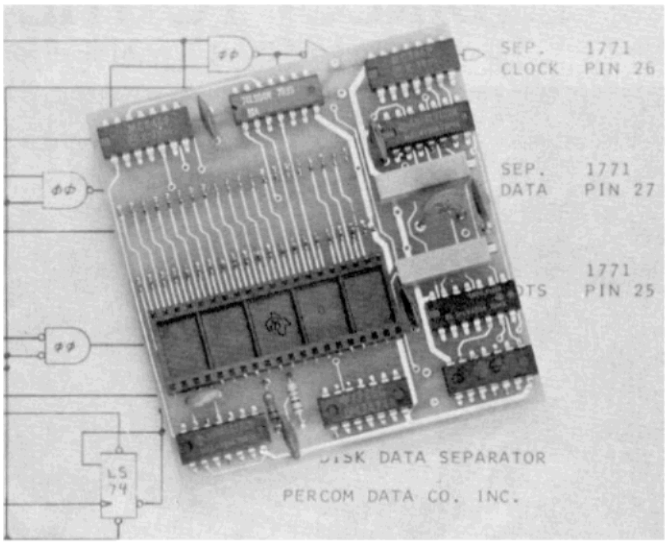
Adapter for TRS-80* computer eliminates disk read errors

Garland, Texas — Harold Mauch, president of Percom Data Company, announced that the company is marketing a simple plug-in adapter for TRS-80* computers that corrects a design deficiency in the disk controller circuit.

The problem, which causes disk read errors, has been traced to Tandy's reliance on a circuit internal to the FD1771 controller IC to perform the function of separating clock and data pulses.

As explained in the *Backgrounder,* use of the internal chip circuit for reliable data-clock separation is a design shortcut which the manufacturer of the controller IC warns against.

The Percom solution, a PC card adapter called the SEPARATOR™, eliminates the problem by substituting an explicit data separator circuit — one which has been used reliably in Percom disk controllers since 1977 — for the internal IC separator circuit.



Percom adapter fixes TRS-80 computer disk controller.*

The SEPARATOR™ is installed without modifying the host system. The user merely removes the FD1771 IC from the host controller, installs the IC in the DIP socket on the SEPARATOR™ card, and plugs the adapter into the vacated socket of the host controller.

Percom cautions that opening the Expansion Interface of the TRS-80* computer, which is required to install the SEPARATOR™, may void the computer's limited 90-day warranty.

The SEPARATOR™, which sells for $29.95, may be purchased from Percom dealers or ordered direct from the factory. The Percom toll-free order number is 1-800-527-1592.

Payment for mail orders may be made by certified check, cashier's check or money order, or charged to a Master Card or VISA account. Texas residents must add 5% sales tax.

Percom Mini-Disk Drives Store More, Cost Less.

Percom mini-disk drives store more data, are more reliable, yet a 40-track Percom drive costs **$100.00 less** than a 35-track Tandy drive.

You can store over 102 Kbytes per disk on Percom TFD-100™ 40-track drives, over 197 Kbytes per disk on TFD-200™ 77-track drives. A patch — supplied free on minidiskette — upgrades TRSDOS* for operation with the newer 40- and 77-track drives.

Both TFD-100™ and TFD-200™ models are available in one-, two- and three-drive configurations.

Prices start at $399 for a single-drive TFD-100™, $675 for a single-drive TFD-200™. Drives are supplied with heavy-duty power supplies. Metal enclosure is finished in compatible silver enamel.

See your nearby Percom dealer or order direct by calling toll-free 1-800-527-1592.

Five-Inch Disks Store More Than Eight-Inch Disks!

Garland, Texas — June 25, 1980 — Percom Data Company has begun production of a double-density disk controller adapter for TRS-80* Model I computers.

Harold Mauch, president of Percom, made that announcement here today, saying that data storage capacity using the adapter and double-density disk operating system — which is included — can be increased to as much as 354 Kbytes per minidiskette.

By comparison, the maximum storage for larger eight-inch disk systems used with the TRS-80* Model I computer is about 290 Kbytes.

Mauch said the PC card adapter, which plugs into the controller chip socket of the computer Expansion Interface, works equally well for either single-density or double-density storage, and users may continue to run programs under TRSDOS*, OS-80™ and other single-density operating systems with the adapter installed.

Price, for the plug-in adapter, the TRSDOS*-like double-density DOS and a utility for converting files and programs from single- to double-density format is $219.95.

BACKGROUNDER

CRC ERROR! TRACK LOCKED OUT!

by the Technical Staff
Percom Data Company

This problem started while we were studying an annoying problem with the TRS-80* computer. Disk drives sold by Percom are realigned and tested before shipment. We noticed, however, that some disk drives would pass the Percom inspection but just would not work reliably on the inner tracks with a TRS-80* computer. These drives were within the manufacturer's specifications, and would function perfectly on other disk systems Percom manufactures — "perfectly" here meaning more than 50 million bytes read without error!

The disk read data separation arrangement in the TRS-80* computer Expansion Interface uses an internal data separator of the FD1771 disk formatter/controller IC. Use of the FD1771 internal data separator is not recommended by Western Digital, the IC manufacturer. The following note appears on page 17 of the FD1771 data sheet:

> Internal data separation may work for some applications. However, for applications requiring high data recovery reliability, WDC recommends external data separation be used.

We suspected the data separator because the problem was most severe on disk inner tracks where storage density is highest and data separation is most critical.

To prove our point, a technician breadboarded a standard Percom data separator circuit, and configured it to plug directly into the FD1771 IC socket of the TRS-80* computer controller.

When connected to the TRS-80* computer, a troublesome drive functioned perfectly! We ran a BACKUP utility many times and never got a track lockout. Before we added the external data separator circuit to the computer, this same drive would always lock out tracks, and would have difficulty reading from the inner (higher number) tracks.

The Percom data separator circuit fixes the mini-disk controller of the TRS-80* computer. The type of drives being used is irrelevant; the circuit eliminates disk read errors resulting from the inability of the Tandy controller design to reliably separate clock and data signals when reading high density inner tracks.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

✓258

PERCOM DATA COMPANY, INC. 211 N. Kirby Street Garland, Texas 75042 (214) 272-3421

™ trademark of Percom Data Company, Inc. *trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

```
5 CLEAR 150
1900 GOSUB 9000
1950 STOP
2000 CLS
2020 GOSUB 9000
2030 X=USR(0)
2040 PRINT X
2050 GOTO 2000
9000 A$="*********1*********2*********3*******
***4*********5*********6*********7*********8***
******9*********0*********1*********2***
******3****"
9020 A1=PEEK(VARPTR(A$)+1)
9030 A2=PEEK(VARPTR(A$)+2)
9040 A=A2*256+A1
9050 POKE 16526,A1:POKE 16527,A2
10000 DATA
      217,221,229,245,175,48,9,1,1,1,
      1,1,1,1,1,1,6,5,79,221
10020 DATA
      42,142,64,221,54,7,45,62,198,211,
      243,62,9,211,243,62,5,211,243,219
10040 DATA
      241,219,242,203,119,40,82,219,242,203,
      71,40,250,219,241,87,203,79,40,3
10060 DATA
      203,2,79,203,87,40,5,62,43,221,
      119,7,203,96,40,9,221,35,221,54
10080 DATA
      7,46,175,87,79,203,10,221,35,219,
      240,221,119,7,219,242,203,95,40,250
10100 DATA
      16,226,203,73,40,6,221,35,221,54,
      7,46,221,35,221,54,7,44,221,229
10120 DATA
      225,62,3,50,175,64,205,188,14,241,
      221,225,217,201
11000 RESTORE
11020 FOR N=0 TO 110
11040 READ D:POKE 40+N,D:NEXT N
11050 RETURN
11060 STOP
```

Program Listing 1. The BASIC program listing. Lines 2000 to 2050 are a sample program in which the voltmeter reading is passed to the variable X.

Therefore, absolute jumps and loads to specific memory locations are not used; all jumps and loads are relative with one significant exception.

The memory location 408EH holds the starting address of the subroutine and is loaded into the IX index register. Since a displacement may be added to the index register, it is possible to set aside space for storing the bytes from the digital voltmeter reading within the string itself.

Second, as mentioned before, some machine codes were avoided. The subroutine is intended to be added to a number of different BASIC programs which need input data from a digital voltmeter. If it produces a string that is difficult to read when displayed or printed, then it is correspondingly difficult to integrate into programs written in BASIC. Such a subroutine will have little value for any scientific work.

Avoiding these codes poses some difficulties in creating subroutines, but we restrict subroutines created in this way to only common ones which are called many times in the execution of a single program. Overcoming the difficulties has its rewards.

The machine language subroutine is shown in Program Listing 2. The listing shows that memory location 72FIH has been chosen as the origin of the program. This selection is completely arbitrary, but some origin must be specified with the TRS-80 Editor/Assembler. This address is never used in the subroutine. If, after assembly, it or other addresses near it should appear in the code, the program contains, incorrectly, absolute jumps or loads.

Steps 100-140. The entry is defined as DVM and, since all of the registers are used, all are initially pushed into the stack or exchanged with the alternate registers. In general, the HL register must be saved since it points to the current cursor position. This register must be restored just before returning to the BASIC program.

Steps 150-210. Step 150 loads zero into the A register and sets the zero flag. The former is essential for storing a zero byte in step 190 in the C register. The latter enables us to use the command JRZ,NEXT which has the opcode 28 09 rather than JR NEXT which has the opcode 18 09.

The ASCII representation for 28H is (, but 18H is a control code which backspaces the cursor and is one that will be avoided where convenient.

Step 170 sets aside space for storing the decoded digits from the voltmeter. While only eight bytes are really needed, the control code for eight is one that is

✓Reader Service—see page 226

also avoided. When the data statement is written with these bytes, the code 01 is entered in each of the nine locations, as can be seen in the data statement 10000 shown in Fig. 2.

Step 180 loads the number of digits that will be read from the voltmeter into register B, while step 190 loads 00 into register C. If only four numbers were to be read, then register B would be loaded with 04.

Step 200 stores the starting address of the program into the index register IX. Whenever USR subroutine is called, the operating system immediately checks memory location 408EH for the beginning address of the subroutine. This is the only absolute memory location used in the program and it is loaded into the index register to determine the start of the buffer storage string.

Based on the number of bytes that have already been written into the program, the first byte of the storage string occurs at IX + 7. This byte must contain the sign of the voltage, initially set negative by step 210.

Steps 220-350. The peripheral interface adapter must now be initialized. Fig. 2 shows the control word that must be used to define operation in mode 1. The control word used in step 280 is 190D, bit pattern 1011 1110. This sets Port A in mode 1 input and Port B in mode 1 input. This control word also has bit 4 high, thereby setting PC6 and PC7 of Port C as input.

In mode 1, the other lines of Port C are committed to the INTR, IBF and STB functions. Therefore, it is immaterial whether bit 0 of the control word is high or low.

Steps 310-320 enable the INTE flip-flop for Port A and steps 330-340 enable the INTE flip-flop for Port B in agreement with the bit SET/RESET control word shown in Fig. 2. These interrupt-enable flip-flops remain enabled during the entire subroutine.

Step 350 is a precaution to ensure that the interrupt request line of Port B has been reset to zero. No other initialization is needed.

Steps 360-390. These steps check to see that the voltmeter is actually turned on. Since the subroutine loops until strobes arrive from the digital voltmeter, the power must be on if the computer is not to loop forever. The initialization procedure set the PC6 line of Port C as an input, and this point is connected to the +5 volt power from the digital voltmeter. The port is read, and the bit compared in step 380. If zero, the subroutine jumps to DVG at the end of the program, restores the registers, and returns to BASIC.

Steps 400-430. The initialization procedure has set the interrupt enable flip-flop of Port B high. A differentiated negative-going pulse signaling that the voltmeter is displaying the first digit, is connected to PC2, the strobe input for Port B. Once this point sees this logical 0 pulse, data is loaded into Ports A and B and, 300 nsec later, IBF is set to a logical 1.

Since STB is differentiated, it returns to a logical 1 very quickly. When STB, IBF, and INTE are all high, then INTR of Port B is set to a logical 1.

In these steps, line PC0 on Port A which contains the logical information about INTR(B) is continually checked. After it is set to a logical 1, the program continues.

Steps 440-500. When the program has reached this point, the digits are read in proper order, but it is necessary to read the sign and to insert the decimal point at the correct place in the string. Although a different order for the subroutine could have been chosen, the 2 millisecond spacing of the digit strobes leaves more than enough time for the program to check the decimal point before reading each digit.

To understand the way in which the decimal point is read, notice the way in which the Keithley voltmeter displays its digital information for volts, amperes and ohms as shown in Table 2.

The simplest way of returning information to the BASIC program is in volts, milliamperes and kilohms. If the voltmeter is reading volts and milliamperes, the decimal point must be

• CONTEST ENDS THIS MONTH • DON'T DELAY •

PRE-CHRISTMAS SALE!

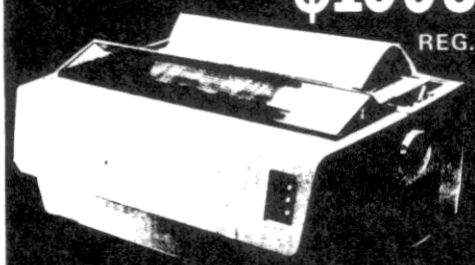
ORDER NOW

NOW OPEN • SHOWROOM AND REPAIR CENTER

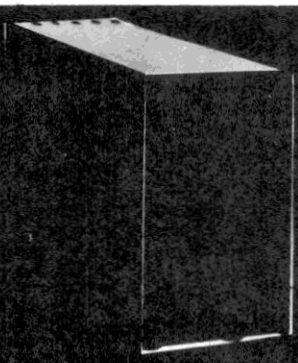
STARWRITER DAISY-WHEEL PRINTER

$1995.00

REG. $2195.00



PARALLEL OR SERIAL INTERFACE
96 CHAR FONT WHEEL
HYTYPE II SUPPLIES
45 CPS (OPTIONAL) ADD $600
TRACTORS (OPTIONAL) $250



77-TRACK DISK DRIVES FOR TRS-80™

$545.00

REG. $ 599.00

ORDER NOW TOLL FREE 1 (800) 345-8102 • VISA-MASTER CHARGE ACCEPTED

6.3 MEG HARD DISK WINCHESTER

FOR TRS-80, APPLE, S-100 $2445.00

SUPERBRAIN™

BY INTERTEC

64K $2995.00

ASK ABOUT OUR SPECIAL PURCHASE GAME SOFTWARE FOR X-MAS.

SOFTWARE	MOD. I	MOD. II
Medical/Dental Patient Accounting		$1500
Word Processing (Magic Wand)		300
General Ledger	$149.95	249
Payroll	99.95	199
Data Base	149.95	299
	Tape	Disk
Upper/Lower Case Modification	$19.95	$24.95
Comprehensive Diagnostics	34.95	34.95
CP/M		$175.00
New DOS + 40 TK		100.00
New DOS/80		145.00

Software Documentation Available • CALL FOR PRICES

DISK DRIVES $ 350 MOD. I TRS-80 COMPATIBLE		
Model II Drives		
1 Drive Single Enclosure		$ 899.00
1 Drive Multiple Enclosure		1069.50
Additional Drives for Mult. Enc.		540.00
Disk Head Cleaning Kit - Mod. I		14.95
Disk Head Cleaning Kit - Mod. II		24.95
4K L II TRS-80		575.70
16K L II		789.60
RS-232		92.10
OK Expansion Interface		278.10
Telephone Modem		179.95
Emulator CRT by Intertec		895.00
CRT Stands	from	139.00
Anti-static Mats		160.00

15% + OFF SELECTED TRS-80'S AND PERIPHERALS

16K LII w/o Key Pads	$652
16K Expansion Interface	355
32K Expansion Interface	450
Centronics 730 Printer	635
Centronics 737 Printer	845
Cable for above Printers	30

PRINTERS

NEC 5510, 5530 w/Tractors	$2950.00
Printer Stands	from 119.00

•••V R DATA'S TRS-80™ SWEEPSTAKES •••

Celebrating V. R. DATA's 8th Anniversary

SWEEPSTAKES RULES

1 ALL ENTRIES MUST BE SUBMITTED ON ORIGINAL ENTRY BLANK
2 ONE ENTRY PER PERSON
3 WINNERS SELECTED BY RANDOM DRAWING. NOTIFIED BY MAIL
4 ENTRIES MUST BE RECEIVED BY 10 31 80
5 VOID WHERE PROHIBITED BY LAW. NO PURCHASE NECESSARY

OVER $1700.00 in PRIZES
GRAND PRIZE - 16K LII TRS-80
TWO SECOND PRIZES - DISK DRIVES
FOUR THIRD PRIZES - $50.00 Gift Certificates

• ABOVE PRICES ARE CASH DISCOUNTED •
CALL FOR OTHER TERMS • DEALER INQUIRES INVITED

MAIL NOW TO ENTER V. R. DATA'S SWEEPSTAKES

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
TELEPHONE _____ OCCUPATION _____
COMPUTER EQUIPMENT OWNED _____
INTENDED USE _____

SEND FREE CATALOG ☐

VR Data
777 HENDERSON BLVD.
FOLCROFT, PA 19032

ORDER TOLL-FREE
1-800-345-8102
LOCAL CALL
(215) 461-5300

• LAST MONTH FOR SWEEPSTAKES • OVER $1700.00 IN PRIZES •

• ENTER NOW! • GRAND PRIZE 16K L II TRS-80™ • EXPIRES 10 31 80

Reader Service—see page 226

moved three places to the left whenever the meter is set to the 0.3, or 200 microamperes, range. For this range, the decimal point must be inserted into the string immediately after the sign byte. However, from Photo 1 the 0.3 volt range on the DVM also corresponds to the 2000 ohm range. From Table 2 it can be seen that the decimal point for the ohms range must be moved one place to the right and inserted as the second, not the first byte, after the sign byte in the string.

Storing in Port B

All of the information about the decimal point is stored in Port B by forcing one of the pins B(4) through B(7) high when the respective 0.2 volt through 200 volt range switch is closed. It is not necessary to connect the 1KV/1AMP/20MEGOHM range switch to the B port. If none of the other range switches have been closed, then this range must be the one that is selected.

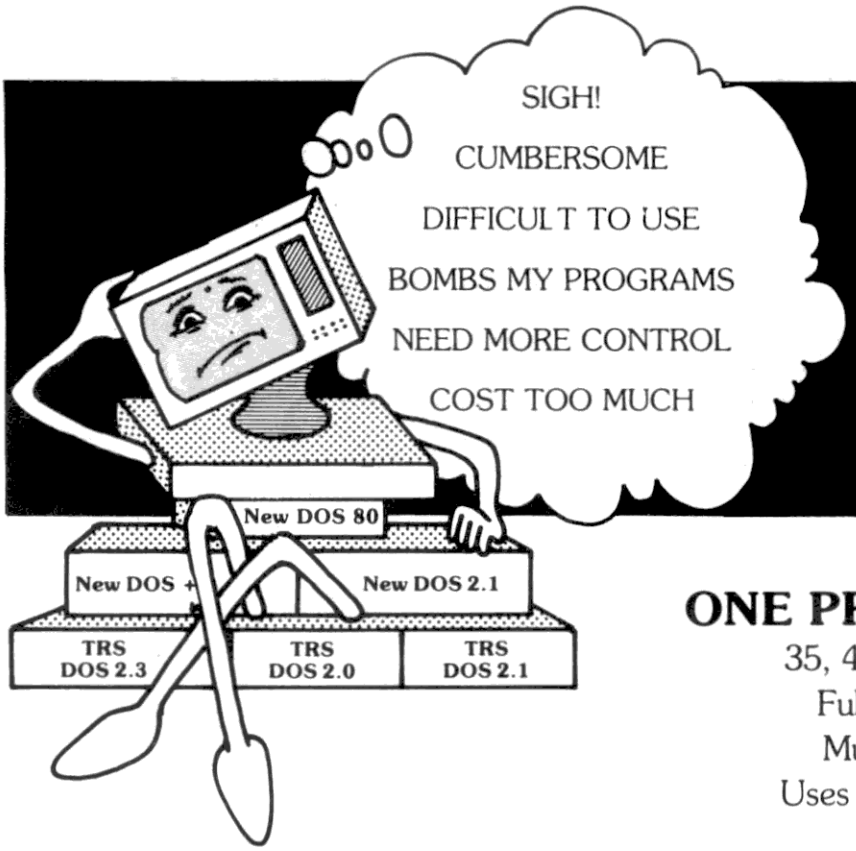
The pin that corresponds to the closing of this switch is B(0) which is connected to the DVM power supply and is, therefore, always high.

Inserting the decimal point at the correct place in the string requires that we check the various bits of Port B, read and stored in register D in steps 450-460. If the voltmeter is set on ohms, pin B(1) is also high. Since an additional shift of the D register to the left is needed when reading ohms (step 490), setting this pin high correctly sets the decimal point for the 20 megohm range. An important part of this routine is step 490.

While the steps through step 730 load in the decimal points correctly for volts and milliamps, the last decimal point, loaded when the voltmeter is set to the 20 megohm range, requires the additional set of steps labeled DVOHM to load it into the string. The C register is set to zero if the decimal point is

0	Avoid. The interpreter thinks it has reached the end of the line.
1-7	No effect. They will not appear when the string is displayed.
8	Backspaces and erases character, but stores correct machine code. When used with the example program, the string is displayed as 9000 X$ = "1237". The characters four, five and six have been erased.
9	Same as the code 1-7.
10	This control code activates the line feed with a carriage return (see control code 26). When used with the example program, the string appears as: 9000 X$ = "1234 5 6 7"
11-13	These move the carriage to the top of the page and have the same effect as 10. They cause no problem except in printing.
14-22	These are identical to 1-7.
23	This converts the display format to 32 characters/line, although it does not appear in the string. It is one of the codes we avoid.
24	This character backspaces the cursor and has the same effect as 8.
25	This advances the cursor. When used with the example program, the string appears as: 9000 X$ = "1234 5 6 7"
26	This code is similar to 10, giving a linefeed, but unlike 10, there is no carriage return. When used with the example program, the string appears as: 9000 X$ = 1234 5 6 7"
27	We avoid this, although the bytes are stored correctly. This control code gives an upward linefeed in contrast to 26 which is a downward linefeed. The listing becomes garbled and difficult to work with.
28	Avoid. Although the bytes are stored correctly, this code homes the cursor creating a listing which is nearly impossible to read.
29	This code is only half bad. When used with the example program, the line number partially disappears and the string appears as 7"00 X$ = "1234. The BASIC interpreter still thinks that this is line 9000 and places it in the program correctly, but depending on its place in the program, this code may create statements that are difficult to renumber or manipulate, particularly if more than one subroutine is used.
30-31	These are the same as 1-7.
32	This is the control code for space. When used with the example program, the string appears as: 9000 X$ = "1234 5 6 7". This code causes no problems.

Table 1. Control Code Summary.



Perk up your TRS-80
—with—
Level IV DOS 4.1
—and—
Level IV Basic (C)
—by—
Vernon B. Hester

ONE PRICE INCLUDES:

35, 40, 77, or 80 Tracks
Full Utilities Package
Multi-Auto Function
Uses Less RAM Memory!

Compatible with all commonly used operative systems!

Get all the "Bells & Whistles
without the glitches & problems!

Only **$119.95**

Limited Time Introductory Offer

$89.95

TRS-80 & New DOS 80 are trademarks of Tandy & Apparat Co's.

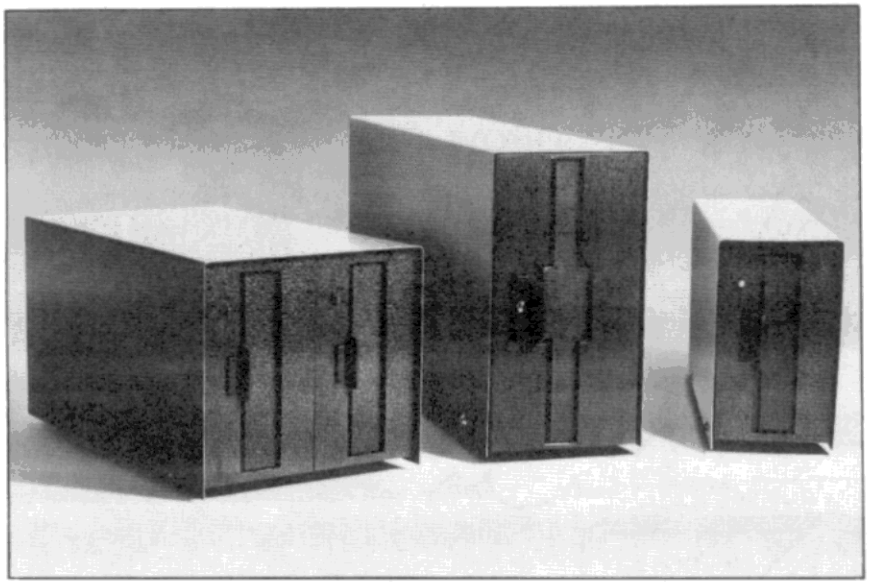
LEVEL IV PRODUCTS, INC.

32238 Schoolcraft Road, Suite F4 • Livonia, MI 48154
313-525-6200 Outside Michigan call 1-800-521-3305

Please add $2.50 for shipping and handling.
COD add another $1.25 plus 25% deposit,
balance certified check or cash.

Dealers Orders Welcome
Open Tues. thru Sat. 11-7 E.S.T.

Reader Service—see page 226



Introducing a new line of cases and power supplies (with or without disk drives!)

The choice is yours

Whether you need professional-looking cases and power supplies for your present hardware or complete disk drive packages, call A.M. Electronics. We manufacture a complete line of affordable, high-quality and *readily-available* disk drive components to expand your TRS-80™ system.

Disk drive enclosures

All of our disk drive enclosure products feature:

- One year power supply guarantee. Each unit is 100% tested, regulated and fused.
- Optimum top-and-bottom venting for improved convection cooling and lower operating temperatures.
- Excellent RF interference shielding. Steel covers come with lustrous dark grey finish to complement your TRS-80™ hardware.
- Built-in provisions for optional extender cable (5¼ only).

Complete disk drive packages

Our complete disk drive packages feature proven MPI and Siemens drives and are designed to meet all your storage requirements. As with everything we sell, A.M. Electronics tests each drive for 100% reliability prior to shipment.

Manufacturer-direct savings

When you order from us, you buy directly from the manufacturer. There are no "middlemen" between us and our customers, so you'll realize significant savings like these:

COMPLETE DISK DRIVE PACKAGES

5¼-inch drives

40-track MPI 51 w/case, power supply & extender cable $350
80-track MPI 91 w/case, power supply & extender cable 560
Dual MPI 51's w/dual case, power supply & extender cables 650
Dual MPI 91's w/dual case, power supply & extender cables 1070

SPECIAL OFFER! 8-inch drives for Model I or II

Single Siemens drive w/case & power supply 695
NEW! Dual Siemens drives w/dual case & power supply 1240

(90 day limited warranty on disk drives)

CASES AND POWER SUPPLIES

5¼-inch enclosures

Single drive unit case & power supply $85
Dual drive unit case & power supply 120

(Extender cables are $15 each extra)

8-inch enclosures

Single drive unit case & power supply 150
NEW! Dual drive unit case & power supply 250

Attention dealers, OEM's & distributors

Having trouble finding cases and power supplies for your disk drives? Call us for details on our attractive pricing and immediate product availability.

™TRS-80 is a trademark of Tandy Corp.

The power behind the drives©

A.M. ELECTRONICS, INC.

3366 Washtenaw Avenue
Ann Arbor, Michigan 48104
(313) 973-2312

✓452

Visit our retail showroom for a "hands on" look at our wide selection of TRS-80™ hardware, software and peripherals. See A.M Electronics, Inc. at the Computer Shows in Chicago, Oct. 16-19 and in Boston, Nov. 20-23.

000.00	.2V/200 a
0.0000	2V/2 ma
00.000	20V/20 ma
000.00	200V/200 ma
0000.0	1kV/1 amp
0.0000	2 kohm
00.000	20 kohm
000.00	200 kohm
0000.0	2 Megohm
00.000	20 Megohm

Table 2. Decimal Point Display.

loaded into one of the first four positions. It is the logical register to use to signal that the voltmeter is set to zero.

In step 500, the accumulator is loaded into the C register. If the voltmeter is set to ohms, then bit 1 must be high, and it will be used in the sequence labeled DVOHM.

Steps 510-550. Having set the decimal point correctly for the possibility of ohms, the program now determines the sign of the number by checking bit 3 or Port B (step 510). If this bit is low, the negative number originally loaded into the string is left there. If bit 3 is high, this byte in the string is replaced with 2BH, which is interpreted as the ASCII character +.

Steps 550-600. In this portion of the subroutine, the decimal point is placed correctly in the string. The process of checking the various bits of Port B is done by checking bit 4 of the D register. If it is high, then the decimal point is read into the string. If it is not high, a digit is read into the string and the D register is rotated to the right. The next time that a digit strobe occurs, bit 4 is again checked to see if the decimal point should be read into the string. Since bit 0 of Port B is wired high, a decimal point must eventually be inserted into the string.

As an example, suppose that the voltmeter is set on the 20 volt range. Bit 6 of Port B and also in the D register is high, and the stored string must read +XX.XXX.

The first time around, the program checks bit 4 of the D register and finds that it is zero. The first digit is read into the string and the D register rotates to the right one place so that bit 5 is

```
72F1            00100          ORG 72F1H
72F1            00110 DVM      EQU $           ;THIS PROGRAM READS A DIGITAL VOLTMETER
72F1 D9         00120          EXX             ;SAVE REGISTERS
72F2 DDE5       00130          PUSH IX
72F4 F5         00140          PUSH AF
72F5 AF         00150          XOR A           ;SET A = 0
72F6 2809       00160          JR Z,NEXT       ;JUMP OVER BUFFER AREA
0009            00170 BUFFER   DEFS 9          ;BUFFER STORES NUMBER STRING
7301 0605       00180 NEXT     LD B,05         ;COMPARISON NUMBER FOR DIGITS
7303 4F         00190          LD C,A          ;STORE 0 IN REGISTER C
7304 DD2A8E40   00200          LD IX,(408EH)   ;STARTING ADDRESS OF PROGRAM
7308 DD36072D   00210          LD (IX+7),2DH   ;LOAD - SIGN INTO FIRST BYTE OF STRING
730C            00220 CTRL     EQU $           ;CONTROL WORDS FOR CLEARS AND ENABLES
                00230                          ;THE 8255 IS BEING OPERATED IN MODE 1
                00240                          ;INPUT A IS PORT 240
                00250                          ;INPUT B IS PORT 241
                00260                          ;INPUT C IS PORT 242
                00270                          ;CONTROL WORD IS 243
730C 3EBE       00280          LD A,190D       ;THESE TWO STEPS INITIALIZE THE PIA,
                00290                          ;  SETTING A & B PORTS AS MODE 1 INPUTS,
730E D3F3       00300          OUT (243D),A    ;  BITS 6 & 7 OF PORT C AS INPUTS.
7310 3E09       00310          LD A,09         ;THESE TWO STEPS ENABLE INTE
7312 D3F3       00320          OUT (243D),A    ;  OF PORT A
7314 3E05       00330          LD A,05         ;THESE TWO STEPS ENABLE INTE
7316 D3F3       00340          OUT (243D),A    ;  OF PORT B
7318 DBF1       00350          IN A,(241D)     ;THIS INSURES THAT PORT B INTR IS LOW
731A            00360 DVON     EQU $           ;CHECK TO SEE IF DVM IS TURNED ON
731A DBF2       00370          IN A,(242D)     ;READ PORT C
731C CB77       00380          BIT 6,A         ;BIT 6 IS CONNECTED TO DVM POWER SUPPLY
731E 2852       00390          JR Z,DVG        ;IF DVM IS OFF, RETURN TO BASIC
7320            00400 DVLOOP   EQU $           ;ROUTINE TO LOOP UNTIL FIRST DIGIT ARRIVES
7320 DBF2       00410          IN A,(242D)     ;BIT 0 ON PORT C GOES HIGH
7322 CB47       00420          BIT 0,A         ;  WHEN FIRST DIGIT ARRIVES
7324 28FA       00430          JR Z,DVLOOP     ;LOOP UNTIL IT ARRIVES
7326            00440 DVA      EQU $           ;CHECK TO SEE IF THE DVM IS SET TO OHMS
7326 DBF1       00450          IN A,(241D)     ;READ PORT B
7328 57         00460          LD D,A          ;STORE PORT B IN REGISTER D
7329 CB4F       00470          BIT 1,A         ;ARE WE READING OHMS?
732B 2803       00480          JR Z,DVB        ;IF NOT, SKIP THE NEXT STEP
732D CB02       00490          RLC D           ;IF OHMS, ADJUST THE DECIMAL POINT
732F 4F         00500          LD C,A          ;IF OHMS, BIT 1 IN C REGISTER WILL BE SET HIGH
7330            00510 DVB      EQU $           ;ROUTINE TO CHECK FOR + SIGN
7330 CB57       00520          BIT 2,A         ;  IS IT + ?
7332 2805       00530          JR Z,DVC        ;IF NEGATIVE, GO TO DVC
7334 3E2B       00540          LD A,2BH        ;IF POSITIVE, FIRST LOAD + INTO ACCUMULATOR
7336 DD7707     00550          LD (IX+7),A     ;  AND THEN INTO THE FIRST BYTE OF THE STRING
7339            00560 DVC      EQU $           ;ROUTINE TO READ DECIMAL POINT
7339 CB62       00570          BIT 4,D         ;CHECK FOR DECIMAL POINT
733B 2809       00580          JR Z,DVD        ;IF NOT THERE, SKIP NEXT 3 STEPS
733D DD23       00590          INC IX          ;POINT IX+7 AT THE NEXT BYTE IN THE STRING
733F DD36072E   00600          LD (IX+7),2EH   ;LOAD DECIMAL POINT INTO STRING
7343 AF         00610          XOR A           ;SET A = 0
7344 57         00620          LD D,A          ;ZERO D AND C IN THESE TWO STEPS
7345 4F         00630          LD C,A          ;  TO AVOID STORING ANY OTHER DECIMAL POINTS
7346            00640 DVD      EQU $           ;ROUTINE TO READ DIGITS INTO STRING
7346 CB0A       00650          RRC D           ;PREPARE TO CHECK DECIMAL POINT NEXT TIME
7348 DD23       00660          INC IX          ;POINT IX+7 AT THE NEXT BYTE IN THE STRING
734A DBF0       00670          IN A,(240D)     ;READ DIGIT
734C DD7707     00680          LD (IX+7),A     ;STORE IT
734F            00690 DVE      EQU $           ;LOOPS UNTIL NEXT DIGIT ARRIVES
734F DBF2       00700          IN A,(242D)     ;READ PORT C
7351 CB5F       00710          BIT 3,A         ;CHECK INTR OF A TO SEE IF DIGIT HAS BEEN LOADED
7353 28FA       00720          JR Z,DVE        ;IF NOT, LOOP UNTIL IT DOES
7355 10E2       00730          DJNZ DVC        ;ONCE IT DOES RETURN TO DVC AND READ IT
7357            00740 DVOHM    EQU $           ;THIS ADDS DECIMAL POINT IF DVM IS SET TO 20 MEGOHMS
```

Program continues

Reader Service—see page 226

```
7359 2806      00760          JR Z,DVLST      ;  THEREFORE, SKIP THE NEXT STEP
735B DD23      00770          INC IX          ;POINT IX+7 AT NEXT BYTE IN STRING
735D DD36072E 00780          LD (IX+7),2EH   ;LOAD DECIMAL POINT INTO STRING
7361           00790 DVLST   EQU $           ;ROUTINE TO READ IN COMMA AT END OF STRING
7361 DD23      00800          INC IX          ;POINT IX+7 AT THE LAST BYTE IN THE STRING
7363 DD36072C 00810          LD (IX+7),2CH   ;LAST BYTE IN STRING MUST BE A COMMA
7367           00820 DVF     EQU $           ;CHANGE STRING TO SINGLE PRECISION NUMBER
7367 DDE5      00830          PUSH IX
7369 E1        00840          POP HL          ;POINT HL AT THE START OF THE STRING
736A 3E03      00850          LD A,03         ;TELL TRS-80 THAT IT IS A STRING
736C 32AF40    00860          LD (40AFH),A    ;  BY LOADING 03 INTO 40AFH
736F CD6C0E    00870          CALL 0E6CH      ;CALL SINGLE PRECISION CONVERSION SUBROUTINE
7372           00880 DVG     EQU $           ;TIME TO RETURN TO BASIC
7372 F1        00890          POP AF          ;RESTORE REGISTERS
7373 DDE1      00900          POP IX
7375 D9        00910          EXX
7376 C9        00920          RET
72F1           00930          END DVM
00000 TOTAL ERRORS
BUFFER 72F8 00170
CTRL   730C 00220
DVA    7326 00440
DVB    7330 00510   00480
DVC    7339 00560   00530 00730
DVD    7346 00640   00580
DVE    734F 00690   00720
DVF    7367 00820
DVG    7372 00880   00390
DVLOOP 7320 00400   00430
DVLST  7361 00790   00760
DVM    72F1 00110   00930
DVOHM  7357 00740
DVON   731A 00360
NEXT   7301 00180   00160
```

Program Listing 2

now high.

The second time around, the program again finds bit 4 low and reads the second digit into the string while rotating the D register to the right once more.

After this rotation, bit 4 in the D register is high and the program reads it into its proper place in the stored string. After reading the decimal point into the string, the index register is incremented so that the next digit does not replace the decimal point. Then steps 600-630 set the D and C registers to zero to avoid the possibility of reading any other decimal point into the string.

Steps 640-730. Before reading the digits, the D register is rotated to the right to set up the next check for the decimal point. The index register is incremented in step 660 and the first digit is finally loaded into the string in steps 670 and 680. Port A contains the four-bit BCD digit from the voltmeter in bits A(0) to A(3).

Since the digit must be stored in the string as an ASCII character, the upper four bits must contain 0011. That is, for example, the ASCII character 34H corresponds to the number 4.

Fig. 1 shows that this has been accomplished by handwiring bits A(4) and A(5) to +5 volts and bits A(6) and A(7) to ground. The remaining digits are accessed after the first digit in sequence. The strobe read by STB(B) occurs only for the first digit.

The differentiated strobe pulses from the remaining digits are ORed using the simple diode OR gate, D1-D5. The output of this gate is read at STB(A).

Steps 700-730 check bit 3 of Port C to determine if INTR(A) has been set high and loop until it is. Once this interrupt request has been set high, it signifies that the data from the next digit has been loaded into Port A. The program jumps back to DVC to check the decimal point and read the digit.

The instruction DJNZ DVC not only executes this jump, but also decrements the B register. Once this register reaches zero, all of the digits have been read and the program proceeds to step 740.

Steps 740-780. If the DVM was set to 20 megohms, then the decimal point has not yet been loaded into the string. Bit 1 of the C register will still be high since step 640 was never executed. This is checked in steps 750 and 760 and, if necessary, the decimal point is loaded into the string in steps 770 and 780.

Steps 790-810. The string must be terminated with either a zero or a comma. Since the appearance of 0 in the string causes the BASIC interpreter to think that the end of the line has been reached, the comma, 2CH, is stored in the string in step 720 as the last byte.

Steps 820-870. The BASIC program can make little use of this string of characters that has been stored in the string. Moreover, Radio Shack does not really tell you how to return the value of the reading to the program. It would be possible to obtain the value by finding the address of the stored string using VARPTR and then PEEKing at the addresses to obtain the value.

But, astonishingly, the single (or double) precision conversion subroutine located in the operating system does all the work. Since only five digits are involved here, single precision is sufficient. It is first necessary to point the HL register to the start of the string in steps 830 and 840. Then, the subroutine needs to know that this is a string, not an integer or double or single precision number, and steps 850 and 860 load the number 03 into memory location 40AFH for this purpose. Step 870 calls the routine at 0E6CH in ROM which converts the string into a single precision number. Were it necessary to convert this string into a double precision number, the program would call 0E65H. The remainder of the subroutine restores the registers and returns to the main BASIC program.

If the subroutine has been called by the statement X = USR(0), then X will equal the single precision value of the number read by the voltmeter. The BASIC program needs only to know that the readings are in volts, milliamperes and kilohms. Reading the Level II manual gives no hint that a single precision number can be returned in this way. If anything, the implication is that only integers can be returned. Nevertheless, a single precision floating point number is returned.

Using the Program

There are two major advantages to the interface as constructed here. The first is that the TRS-80 truly becomes a scientific instrument capable of storing and analyzing data, while allowing programs to be written in a high level language.

By imbedding the subroutine into BASIC, the major portion of the program can be written in BASIC which has all of the virtues of a high level language in the ease of displaying data, writing strings, and creating programs.

The second advantage is related to the first. By writing the subroutine as a string in a BASIC statement and storing the entire program on disk or tape, it may be accessed at any time for use by another program that you may be writing.

A library of subroutines such as the one described here may be prepared not only for reading data from digital voltmeters, but also for reading analog-to-digital converters, executing visual displays or even performing mathematical iterations. ■

WE WILL NOT BE UNDERSOLD

16K MEMORY UPGRADE KITS 2 for $85 $45

for TRS-80*, Apple II, (specify): **Jumpers** $2.50

PRINTERS NEC Spinwriter

Letter Quality High Speed Printer
Includes TRS-80* interface software, quick change print fonts, 55 cps, bidirectional, high resolution plotting, graphing, proportional spacing: R.O. $2579
R.O. with Tractor Feed **$2679** KSR with Tractor Feed $2995

779 CENTRONICS TRACTOR FEED PRINTER $969
Same as Radio Shack line printer I

737 CENTRONICS FRICTION & PIN FEED PRINTER $799
9 × 7 matrix

730 CENTRONICS FRICTION & PIN FEED PRINTER $629
7 × 7 matrix Same as Radio Shack line printer II

P1 CENTRONICS PRINTER $269
Same as Radio Shack quick printer

PAPER TIGER (IP440) $939
Includes 2K buffer and graphics option

TI-810 Faster than Radio Shack line printer III
Parallel and serial w/TRS-80* interface software with upper and lower case and paper tray $1599

OKIDATA Microline 80 Friction and pin feed $559
Tractor Feed, friction, and pin feed $679

EATON LRC 7000 + 64 columns, plain paper $299

ANADEX DP-9500 $1359 DP-8000 $825

DISK OPERATING SYSTEMS

PATCHPAK #4 by Percom Data			$ 8.95
CP/M® for Model I, Zenith	**$145** •	for Model II, Altos	$169.00
NEWDOS Plus		**40 track**	$ 99.00
NEWDOS 80			$135.00

ACCESSORIES

HEAD CLEANING DISKETTE: Cleans drive Read/Write head in 30 seconds. Diskettes absorb loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use. Specify 5¼" or 8". **$20 ea/$45 for 3**

FLOPPY SAVER: Protection for center holes of 5¼" floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool, tough 7-mil mylar reinforcing rings. Installation tools and rings for 25 diskettes. $11.95
Re-orders of rings only $ 7.95

EXTERNAL DATA SEPARATOR: Eliminates data separation problems (crc). Improves reliability. This plug in unit comes fully assembled and tested. $29.95

RS232 $84.00

DISK-DRIVE EXTENDER CABLES: Fits all mini-disk drives. $16.95

SIX (6) PRONG ISOLATOR $54.00

AC FILTER/6 PRONG POWER STRIP $39.00

DISK DRIVE CABLES: 2 drive **$29.00** 4 drive $35.00

DUST COVERS: TRS-80/Apple $ 7.95

PLASTIC DISKETTE HOLDER $ 8.00

RF MODULATOR: Adapts video to TV $35.00

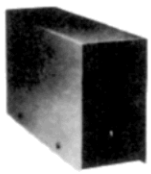
TRS-80 & OTHER MYSTERIES $18.95

CAT MODEM Originate and answer same as Radio Shack Telephone Interface II $148

LEEDEX MONITOR Video 100 $119

ZENITH Color Monitor $379

SANYO Model VM 4509 Monitor $179



DISK DRIVES $314

40 track, 102K Bytes. Fully assembled and tested. Ready to plug-in and run the moment you receive it. Can be intermixed with each other and Radio Shack drive on same cable. TRS-80* compatible silver enclosure. 90 day warranty. One year on power supply. **External card edge included.**

FOR TRS-80*

CCI-100	5¼", 40 Track (102K Bytes) for Model I	**$314**
CCI-280	5¼", 80 Track (204K Bytes) for Model I	$549
CCI-800	8" Drive for Model II (½ Meg Bytes)	$795
For Zenith Z89		
CCI-189	5¼", 40 Track (102K Bytes) add-on drive	$394
Z-87	Dual 5¼" add-on drive system	$995
DISKETTES —	Box of 10 (5¼") - with plastic library case	$24
	8" double density for Model II (box of 10)	$36

COMPLETE SYSTEMS

ALTOS 64K, DD, SS, 2-Drive, 1MB $3995

TRS-80* Model II-64K $3499

TRS-80* LEVEL II-16K with keypad $689

TRS-80* Expansion Interface $249

APPLE 16K $989

HEWLETT PACKARD HP-85 $2999

ZENITH Z89, 48K all-in-one computer $2555

ZENITH Z19 $740

TELEVIDEO 912B $745 920B $769

ATARI 400 $489 **ATARI 800** $769

APF M1000 $99 **IM-1** $499

MATTEL INTELLIVISION $249

SOFTWARE FOR THE TRS-80* Software w/Manual

INTELLIGENT TERMINAL SYSTEM ST-80-III BY LANCE MIKLUS: Enables a TRS-80* to act as a dial-up terminal on any standard time sharing network. Provides a TRS-80* with control key, ESC Key, Repeat Key, Rub Out Key, Break Key, full upper and lower case support, selectable printer output and program selectable transmission rates. **$139**

CCA-DATA MANAGEMENT SYSTEM: Automate your information processing tasks. You can create a file of customer information; quickly and easily add, delete or update records; search a file; keep a file in order of the value in any field; and print records and labels in any desired sequence or from just a part of a file. Requires 32K TRS-80 and one drive. **$74.95**

S & M SYSTEMS
INSEQ-80™: Indexed Sequential Access Method (ISAM) for the TRS-80 Model I. A must for anyone writing business programs. Eliminate wasted disk space from direct record processing. Split second access to any record. Access data records instantly via alpha/numeric "key" eg. Part NR, zip code or sequentially in ascending key sequence. Add/modify records in any order. Access up to three files per program—Files may be spread over multiple disks. Machine language processing from your basic program. Utility program to convert direct files to INSEQ-80 format. **$49.95**

FULLY INTERACTIVE ACCOUNTING PACKAGE: ISAM (INSEQ-80) based. Includes General Ledger, Accounts Payable, Accounts Receivable and Payroll. System runs "stand alone" or "co-ordinated G/L" at users option. Based on Osborne accounting method. Requires 32K, TRS-80, 2 or 3 drives. N/A CA.

General Ledger	**$99**
Accounts Receivable	**$99**
Accounts Payable	**$99**
Payroll	**$99**

Osborne books: Req'd as additional documentation $20 ea

INVENTORY Requires 32K, TRS-80, 1 drive **$125**
INSORT-80: Callable form BASIC via USR. Sorts "Random" Disk Files. "Disk" to "Disk" sort times — 350 records in 35 secs, 1000 records in 6 minutes, 3500 records in 12 minutes. Machine language processing. Up to 35 sort keys ascending/descending. Utility to build BASIC program. Runs under NEWDOS. **$49.95**

CP/M BASED SOFTWARE for Zenith, Altos, Radio Shack, Apple Software w/Manual

Z-80 SOFTCARD FOR APPLE: Your key to future software expansion. Get the best of both worlds, Apple's 6502 and CP/M Z-80. Plug in the card and get a Z80. Supports Apple language card and all Apple peripherals. Comes with set of three manuals. **$339**

CCI-TELNET VERSION 5: A communication Package which enables microcomputer users to communicate both with Large Mainframes and other microcomputers. Extensive commands make it useful in many applications where communication between computers is necessary. Powerful terminal mode enabling user to save all data from a session on disk. Completely CP/M compatible. Multiple communication protocols supported. Able to transfer files in both directions without protocol where the other machine does not support any protocol. Extensive ON-SCREEN help. Source code provided. **$149**

MICROPRO-WORD-STAR: Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace. Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning. **$399**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP

✓298

TO ORDER CALL TOLL FREE 1-800-343-6522
Massachusetts residents call **(617) 242-3361**
For detailed technical information, call 617/242-3361
Hours: 10AM-6PM (EST) M-F (Sat. till 5)

5 Dexter Row, Dept. M10M
Charlestown, Massachusetts 02129

Massachusetts residents add 5% sales tax

Quantities on some items are limited

VISA*

master charge

*TRS-80 is a Tandy Corporation Trademark ®Digital Research

A residential heating and insulation analysis.

Cold Comfort

Dan Keen
Dan Laughlin
Rd 1, Box 432
State Highway 83
Cape May Courthouse, NJ 08210

Armed with a few facts about your home, this program for the Level II will determine how much money you spend a year on heating, what the size of your heater should be, how much you can save if you add more insulation and how many years until the cost of the additional insulation pays for itself. It is a simple program with fairly accurate results.

Using the Program

First the programs ask the delta temperature in Fahrenheit for your area, which is the average maximum difference between inside and outside temperatures.

Next, you are asked to input therms. To arrive at this figure you must use the chart which is displayed on the screen and in-

```
FIRST ANNUAL COST = $1800.27
NEW ANNUAL COST = $1701.74
ANNUAL SAVINGS = $98.52
WHAT WAS THE COST OF ADDITIONAL INSULATION? 300.00
IT WILL TAKE 3.045 YEARS TO PAYBACK THE COST.
```

Sample Listing 2.

```
ENTER TEMPERATURE DIFFERENCE (IN DEGREES FAHRENHEIT)? 60
ENTER THERMS (FROM CHART) FOR YEAR IN 100,000 BTU'S? 2
ENTER FLOOR AREA (IN SQ. FT.)? 1200
ENTER FLOOR R? 11
MAXIMUM HEAT LOAD FOR FLOOR IS 6545.40 BTU'S/HOUR.
ANNUAL HEAT LOAD FOR FLOOR IS 218.182 THERMS/YEAR.
ENTER CEILING AREA (IN SQ. FT.)? 1200
ENTER CEILING R? 19
MAXIMUM HEAT LOAD FOR CEILING IS 3789.47 BTU'S/HOUR.
ANNUAL HEAT LOAD FOR CEILING IS 126.316 THERMS/YEAR.
ENTER WALL AREA (IN SQ. FT.)? 1220
ENTER WALL R? 11
MAXIMUM HEAT LOAD FOR WALL IS 6654.55 BTU'S/HOUR.
ANNUAL HEAT LOAD FOR WALL IS 221.818 THERMS/YEAR.
ENTER WINDOW AREA (IN SQ. FT.)? 120
ENTER WINDOW R? 1
MAXIMUM HEAT LOAD FOR WINDOWS IN 7200 BTU'S/HOUR.
ANNUAL HEAT LOAD FOR WINDOWS IS 240 THERMS/YEAR.
ENTER DOOR AREA (IN SQ. FT.)? 40
ENTER DOOR R? 2
MAXIMUM HEAT LOAD FOR DOOR IS 42 BTU'S/HOUR.
ANNUAL HEAT LOAD FOR DOOR IS 140 THERMS/YEAR.
TOTAL HEAT LOSS = 27928.4 BTU'S/HOUR.
TOTAL ANNUAL HEAT LOSS = 930.947 THERMS/YEAR.
HIT ENTER TO FIND ANNUAL COSTS?
WHAT TYPE OF FUEL DO YOU HAVE? (O)IL (E)LECTRIC (G)AS E
COST OF KWH OF ELECTRIC? .066
ANNUAL COST = $1800.27
HIT ENTER TO DETERMINE PAYBACK TIME IF MORE INSULATION IS ADDED?
```

Sample Listing 1.

Program Listing 1.

```
0 CLS:PRINT@13,"CALCULATING INSULATION PROFIT MARGINS
50 MS="$$#,###.##"
100 INPUT"ENTER TEMPERATURE DIFFERENCE (IN DEGREES FAHRENHEIT)";T
110 CLS:PRINT"ENTER THERMS (FROM CHART) FOR YEAR IN 100,000 BTU'S";:
115 GOSUB2000
116 PRINT@53,"";:INPUT TH
118 CLS
120 INPUT"ENTER FLOOR AREA (IN SQ. FT.)";A
130 INPUT"ENTER FLOOR  R ";R
140 GOSUB1000
150 PRINT"MAXIMUM HEAT LOAD FOR FLOOR =";H;" BTU'S/HOUR "
160 PRINT"ANNUAL HEAT LOAD FOR FLOOR =";AH;" THERMS/YEAR "
170 H1=H:A1=AH
180 PRINT:INPUT"ENTER CEILING AREA (IN SQ. FT.)";A
190 INPUT"ENTER CEILING  R ";R
200 GOSUB1000
210 PRINT"MAXIMUM HEAT LOAD FOR CEILING =";H;" BTU'S/HOUR "
```

Program continues

INTERACTIVE FICTION

The Literature of the Future—Here Today

If you like fiction you'll love Interactive Fiction—tales told on a computer, so that you, the reader, become a character in the story!

For TRS-80 with DISK and 32K
(TRS-80 is a trademark of Tandy Corp)

Six Micro-Stories
A good introduction to Interactive Fiction. Become a financial tycoon, a WW-II spy, and more $15.00

His Majesty's Ship "Impetuous"
You are the captain of a ship-of-the-line in the glorious days of fighting sail $20.00

Two Heads of the Coin
You are confronted with a psychological mystery that taxes your imagination to the utmost $20.00

INTERACTIVE FICTION
BOX 603
INVERNESS CA 94937
✓429

(Calif. residents add 6% tax)

INTELLIGENT TERMINAL

Use your TRS-80* as a dial-up terminal on any standard timeshare system. Includes control keys. Print command lists data on printer. Store command writes data to disk file. Write command reads disk file and outputs to RS232 interface $49.95

ADDITIONAL SOFTWARE AVAILABLE:

ACCOUNTS RECEIVABLE **$495**
Maximum 9000 customers with up to 3000 transactions per customer per billing period. Includes customer status reports, past due billings, mail list, etc. Reports may be run at any time as often as desired. High-quality, professional software.

GENERAL LEDGER **$495**
For medium sized business. Designed and proven impossible to unbalance books. IRS auditable. Up to 9 departments. High-quality, professional software.

PAYROLL **$495**
For up to 120 people per year. Same high-quality, professional software.

MAIL LIST **$9.95**
Basic, no-frills program. Easy to use. Lists all (or range of) addresses. Prints 4 lines by 30 chars on standard 3½" x 15/16" label.

UNILOGIC ✓423
P.O. Box 160
PARIS, KY 40361
(606) 987-2678
(606) 987-4310
(32K TRS-80* with disk drive required.)
*TRS-80 is a trade mark of Tandy Corp.

$250?

Our 50/80 INTERFACE lets you connect your Radio Shack TRS 80 - Model 1 to the IBM Electronic Typewriter Mod 50, 60 or 75. Aside from yielding the best looking printouts and listings you'll ever see, our system lets you center titles, underline words & phrases, justify numerical columns, indent text and more - from your own programs and most others. Right justified proportionally spaced typesetting is even possible with the IBM50! The most cost effective word processor ever. (this ad is an example) $250.00

SOFTWARE also available for any TRS80:

SUPERDIR - displays a menu like directory in DOS from which you can RUN or KILL any program, display updated FREE space & print the display, all with single key commands. In fast acting Z80 code, only $14.95 (DOS).

INMOD3 - easily used Z80 system program that can give any BASIC program professional keyboard entry. Blinking cursor, upper/lower case, user defined input length, repeat keys & single keystroke control codes. Makes INKEY$ obsolete; saves 1000 bytes over BASIC equivalent; for the rankest amateur! $14.95 (L2 or DOS).
INMOD3 Plus - same but works with Percom "Speak 2 Me": each character is spoken as entered! $19.95

* * * INTRODUCING our new MX80 FIRMWARE Interface and Modules. Software now in hardware form; utilizes the unused 2K lower mem. Write or call. VISA & MC. Dealers encouraged!

MEDIAMIX
PO Box 8775
Universal City, CA. 91608
213-475-9949
Micronet# 70250,321

THE BOOKKEEPERS

FOR INFO CALL (603)-447-2745

Full Charge Bookkeeper—48K, 3 DRIVE, w/ALPHA **$199.95**
Intermediate Bookkeeper—48K, 2DRIVE & Printer **$189.95**
Cheap Bookkeeper—32K, 2DRIVE & Printer **$175.00**

All Above Are Daily Journal—G/L Systems

Hex Code Converter, Loan Payment Finder, & Amortization Table, 16K, 1DRIVE & Printer—ALL 3 $29.95

STURDIVANT & DUNN, INC. ✓82
BOX 277, 124 WASHINGTON ST., CONWAY, NH, 03818

If you're serious about the stock market, you need Tickertec™

Watch 48 to 400 of your favorite stocks without a 15 minute delay.

Tickertec™ is a computer program that displays the NYSE or AMEX tickertape on your TRS-80™ Model I or both exchanges as an option on the Model II. You see every trade as it is reported by the exchange and track the last ten trades, tickertape reported volume, and high and low limits on the stocks you are watching. Tickertec program prices start at $1,000.00 with many optional features available including hard copy and portfolio management systems. Programs may be purchased for cash (i.e., hard dollars) or payment can be arranged in the form of discounted brokerage commissions (i.e., Soft Dollar Software™). Exchange fees are extra. Call for FREE brochure TOLL-FREE at **(800) 223-6642**; in New York call **(212) 687-0705**; or circle the reader service number.

MaxUle& Company Inc. ✓428
6 East 43rd Street, N.Y., N.Y. 10017

Please note: Our CRT SCREENS have been purchased by thousands of individuals, the Department of the Navy, several government agencies, and dozens of the country's top corporations and universities.

- Give your CRT the luminous green characters found on the very expensive computer systems.
- Add a professional look to your system and your programs.
- Dramatically improved contrast for easier reading and improved graphics.

We manufacture an optically correct, 1/8" plexiglas* screen that mounts easily over the CRT on your video monitor. This is a quality accessory that enables your TRS-80* monitor to produce the luminous green characters identical to those found on expensive terminals. For business applications this means enhanced appearance and reduced eye strain, for the hobbyist, graphics are brighter and bolder. The screen may be easily removed -- no modification to monitor.

Screen for Model I $19.95
Screen for Model II . . . $24.95
VISA - Mastercharge

We ship within 24 hours
30-day money back guarantee

✓142

National Tricor, Inc. / 3335 Greenleaf Blvd., Kalamazoo, MI 49008 / 616-375-7519

```
220 PRINT"ANNUAL HEAT LOAD FOR CEILING =";AH;" THERMS/YEAR."
230 H2=H:A2=AH
240 PRINT:INPUT"ENTER WALL AREA (IN SQ. FT.)";A
250 INPUT"ENTER WALL  R  ";R
260 GOSUB1000
270 PRINT"MAXIMUM HEAT LOAD FOR WALLS =";H;" BTU'S/HOUR.":PRINT"ANNUAL HEAT LOAD FOR WALL =";AH;" THERMS
    /YEAR."
280 H3=H:A3=AH
290 PRINT:INPUT"ENTER WINDOW AREA (IN SQ. FT.)";A
300 INPUT"ENTER WINDOW R";R
310 GOSUB 1000
320 PRINT"MAXIMUM HEAT LOAD FOR WINDOWS =";H;" BTU'S/HOUR.":PRINT"ANNUAL HEAT LOAD FOR WINDOWS =";AH;"
    THERMS/YEAR."
330 H4=H:A4=AH
340 PRINT:INPUT"ENTER DOOR AREA (IN SQ. FT.)";A:INPUT"ENTER DOOR  R";R
350 GOSUB1000
360 PRINT"MAXIMUM HEAT LOAD FOR DOOR =";H;" BTU'S/HOUR.":PRINT"ANNUAL HEAT LOAD FOR DOOR =";AH;" THERMS
    /YEAR."
370 H5=H:A5=AH
380 H=H1+H2+H3+H4+H5:AH=A1+A2+A3+A4+A5
390 H=H+(.1*H):AH=AH+(.1*AH)
400 PRINT"TOTAL HEAT LOSS =";H;" BTU'S/HOUR.":PRINT"TOTAL ANNUAL HEAT LOSS =";AH;" THERMS/YEAR."
410 PRINT:INPUT"HIT ENTER TO FIND ANNUAL COSTS";A$
420 CLS:PRINT"WHAT TYPE OF FUEL DO YOU HAVE?"
430 PRINT"(O)IL
435 PRINT"(E)LECTRIC
440 PRINT"(G)AS
442 PRINT@31,CHR$(143);:FORQ=1TO250:NEXT:PRINT@31," ";:FORQ=1TO250:NEXT
450 Z$=INKEY$:IFZ$=""GOTO442
455 IF Z$="O" GOTO 500
456 IF Z$="E" GOTO 600
457 IF Z$="G" GOTO 700
459 GOTO420
460 ON2GOTO500,600,700
500 CLS:INPUT"COST OF #2 FUEL OIL PER GALLON";C
510 AC=AH*C*.71
520 PRINT"ANNUAL COST=";:PRINTUSING"$";AC:GOTO800
600 CLS:INPUT"COST OF KWH OF ELECTRIC =";C
610 AC=AH*C*29.3
620 PRINT"ANNUAL COST=";:PRINTUSING"$";AC:GOTO800
700 CLS:INPUT"COST FOR ONE CUBIC FOOT GAS =";C
710 AC=AH*C*100
720 PRINT"ANNUAL COST=";:PRINTUSING"$";AC
800 PRINT"HIT ENTER TO DETERMINE PAYBACK TIME IF MORE INSULATION ADDED":INPUTZ$
805 E=E+1:IFE=>3END
806 IFE=2THENGOTO830
810 CLS:ZZ=AC
820 IFE=1GOTO100
830 CLS:PRINT"FIRST ANNUAL COST =";:PRINTUSING"$";ZZ
840 PRINT"NEW ANNUAL COST =";:PRINTUSING"$";AC
850 IFAC>ZZPRINT"NEW COST IS MORE! DON'T DO IT!":END
860 NC=ZZ-AC:PRINT"ANNUAL SAVING =";:PRINTUSING"$";NC
870 PRINT:INPUT"WHAT WAS THE COST OF ADDITIONAL INSULATION";AI
890 PB=AI/NC
900 PRINT"IT WILL TAKE ";PB;" YEARS TO PAYBACK THE COST."
```

Program continues

put the efficiency of your heater and the number of degree days. Your meter man should have both of these figures.

Locate the number of degree days on the Y-axis of the chart. Now move to the right until you hit the curve with the percent efficiency of your heater. Note the number on the X-axis (from 1 to 8) and use it to ENTER THERMS.

The area and thermal resistance (R) of the floors, walls, windows, doors and ceilings are entered. Typical R values are: 11 for 3½-inch standard fiberglass insulation; 19 for 6-inch standard fiberglass insulation; two for storm or insulated windows and doors. Exact values should be obtained from local building suppliers.

To account for cracks and openings that allow warm air to escape, heat load is calculated using 10 percent for the infiltration factor. It is assumed that this is a standard house of average quality construction, with evenly dispersed windows and no solar aid. The heat load figure represents how much heat you must put into the home to maintain the delta temperature inside.

To find the annual heating cost, you are asked what kind of heat you have. Type O, E or G, but do not hit ENTER. Respond to COST OF FUEL in dollars, such as .85 rather then 85¢ per gallon.

To computer the money saved by adding insulation and the number of years for the investment to pay for itself, the program will again ask you areas and R values. This time you will enter adjusted R numbers at the locations where you want to add extra insulation.

Modification

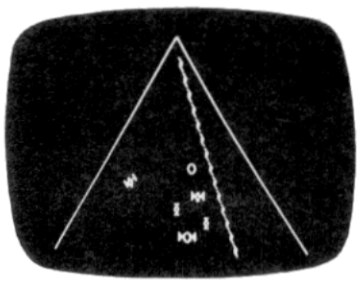
A sample run follows. The chart is not shown which appears in the program at ENTER THERMS.

In Sample Listing 2 the computer again asks for areas and thermal resistances for ceiling, floors, etc. This time let's change the R value in the ceiling to 30 to represent an addition of insulation. ■

FROM PROGRAMMA
HI-RESOLUTION GRAPHICS FOR THE TRS-80®

LOWER CASE
The 80-GRAFIX board includes two sets of lower case characters at no additional cost.

INVERSE VIDEO
The 80-GRAFIX board allows you to do inverse video to high-light your screen displays.



DEMONSTRATION PROGRAMS
The 80-GRAFIX board is supplied with a Character Generator software and several demonstration programs.

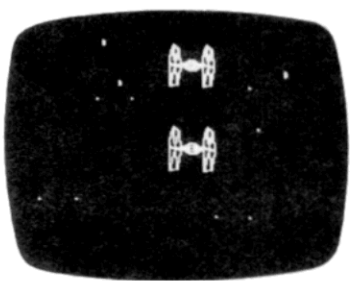
FINALLY, AT LAST. . .

HI- RESOLUTION GRAPHICS is available for your TRS-80 computer system. The 80-GRAFIX board from PROGRAMMA International, Inc. gives your TRS-80 high resolution capability that is greater than the Commodore CBM/PET or even the revered APPLE II.

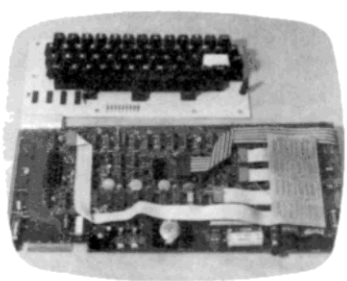
80-GRAFIX gives the TRS-80 an effective screen of 384X192 pixels, versus the normal 127X192 for the TRS-80, 80X50 for the CBM/PET, or the 280X192 of an APPLE II. As an added feature, 80-GRAFIX offers you lower case characters at no additional cost. Of course, you can also create your own set of up to 64 original characters using the supplied Character Generator software.

The 80-GRAFIX board is simple to install (note that this voids your Radio Shack warranty), and programming is done through BASIC. 80-GRAFIX opens up a whole new realm of software development and excitement never dreamed of for the TRS-80!

CHARACTER GENERATOR
The supplied character generator software allows you to create your own character set of up to 64 original characters.

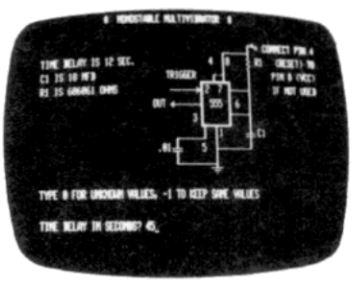


REAL-TIME GRAPHIC GAMES
With the 80-GRAFIX board you can write exciting real-time games using BASIC.

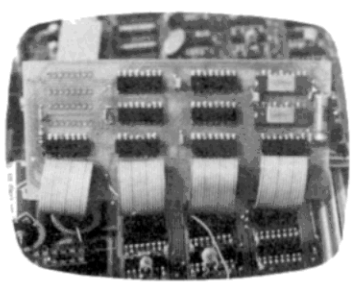


EASY INSTALLATION
The 80-GRAFIX board is simple to install and fits inside the TRS-80 case.

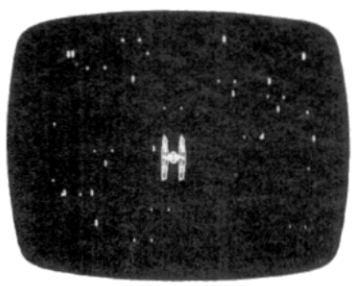
GRAPHICS GALORE
The 80-GRAFIX board and the supplied Character Generator allow you to become an artist.



ELECTRONIC DESIGN
The 80-GRAFIX board has unlimited application in Electronic design and Education.



80-GRAFIX HI-RESOLUTION
Finally, the only means to protect your computer investment is to order an 80-GRAFIX board TODAY!



EXCITEMENT & FUN
Open up a new realm of software development with the 80-GRAFIX board.

Available exclusively through PROGRAMMA at the cost of $149.95
Please check with us for availability prior to ordering
VISA and MASTERCHARGE accepted
TRS-80 is a registered trademark of the Tandy Corp.

PROGRAMMA INTERNATIONAL, INC.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579 · 384-1116 · 384-1117

```
910 END
1000 H=(A*T)/R:AH=(TH*A)/R:RETURN
2000 FORX=6TO7:FORY=13TO44:SET(X,Y):NEXTY,X
2010 FORX=6TO55:SET(X,44):NEXT
2020 G$="  !  !  !  !  !  !  !  !  !  !"
2030 PRINT@260,G$;:PRINT@324,G$;:PRINT@388,G$;:PRINT@452,G$;:PRINT@516,G$;:PRINT@580,G$;
2040 PRINT@644,G$;:PRINT@708,G$;:PRINT@772,G$;:PRINT@836,G$;
2050 PRINT@966,"1  2  3  4  5  6  7  8";
2060 LL=256
2070 XX=10
2080 FORQ=1TO10:PRINT@LL,XX;:LL=LL+64:XX=XX-1:NEXT
2090 SET(37,40):SET(38,40):SET(39,39):SET(40,39):SET(41,38):SET(42,38):SET(42,38):SET(43,38):SET(43,37):SET(44,37):FORX=45TO47:SET(X
,36):NEXT
2100 FORX=48TO51:SET(X,35):NEXT:FORX=52TO55:SET(X,34):NEXT:PRINT@733,"10%";
2110 X=18:Y=40:FORQ=1TO7:SET(X,Y):SET(X+1,Y):X=X+2:Y=Y-1:NEXT
2120 FORX=30TO33:SET(X,34):NEXT:SET(34,33):SET(35,33):SET(36,32):SET(37,32):FORX=38TO41:SET(X,31):NEXT:SET(42,30):SET(43,30)
2130 SET(44,29):SET(45,29):FORX=46TO49:SET(X,28):NEXT:SET(50,27):SET(51,27):FORX=52TO57:SET(X,26):NEXT:PRINT@541,"20%";
2140 X=36:Y=24
2150 FORQ=1TO12:SET(X,Y):X=X+1:Y=Y-1:NEXT:SET(35,24):SET(34,25):SET(32,26):SET(33,26):SET(31,27):SET(30,28):SET(31,28):SET(28,29):SE
T(27,30):SET(26,30):SET(25,32):SET(24,32):SET(22,33):SET(21,34):SET(20,35):PRINT@210,"60%";
2160 X=34:Y=13:FORQ=1TO4:SET(X,Y):SET(X,Y+1):X=X-1:Y=Y+2:NEXT:X=31:Y=22:FORQ=1TO15:SET(X,Y):X=X-1:Y=Y+1:NEXT:PRINT@217,"40%";
2170 SET(28,13):X=28:Y=14:FORQ=1TO8:SET(X,Y):X=X-1:Y=Y+1:NEXT:X=21:Y=21:FORQ=1TO7:SET(X,Y):SET(X,Y+1):Y=Y+2:X=X-1:NEXT:PRINT@206,"80
%";
2180 X=26:Y=13:FORQ=1TO8:SET(X,Y):X=X-1:Y=Y+1:NEXT:X=19:Y=20:FORQ=1TO10:SET(X,Y):SET(X,Y+1):X=X-1:Y=Y+2:NEXT:PRINT@201,"100%";
2190 PRINT@481,"DEGREE [          HEATER";:PRINT@545,"DAYS   [          EFFICIENCY";:PRINT@609,"IN     [";:PRINT@673,"1000'S [";:
2200 PRINT@744,STRING$(22,94);:PRINT@808,"THERMS (100,000) BTU'S";:PRINT@872,"  PER SQ FT PER YEAR";:
2210 X=34:Y=32:FORQ=1TO12:SET(X,Y):X=X+1:Y=Y-1:NEXT:RETURN
```

An inexpensive hardware mod for those who want both Level I and II capability.

Two BASICs Are Better Than One

Allen W. Erickson
13868 Far Hills Lane
Dallas, TX 75240

If you are contemplating an upgrade of your TRS-80 Level I to Level II BASIC, you're probably also wondering what to do about all the Level I programs you've accumulated or written.

If you already own a TRS-80 Level II, you may be wondering how you can run Level I programs on your machine. A quick look through *80 Microcomputing* and other personal computing magazines reveals a great many Level I programs that have not yet been converted to Level II and perhaps never will.

You can rewrite the programs yourself, but that's too much work. Besides, you may not have the source listings. And it's not very practical to change the Level I and Level II ROM chips every time you change programming languages.

The Switch Between

The obvious solution to these problems is to have both Level I and Level II BASIC installed in your machine, with some means to switch between them. Such modifications have been published before, but in every case I've seen, the mod required printed circuit trace cutting, piggyback components, wire unsoldering and resoldering, etc.

The modification described here requires absolutely no alterations to the CPU board and, if you have the single-chip BASIC I ROM, less than five dollars in parts.

A note of caution: *This mod will work only with the single-chip BASIC I ROM and CPU boards with separate chip select lines to pin 20 of the two ROM sockets (Z33 and Z34). Some early TRS-80s may not have separate lines. If in doubt, check with your dealer.*

If you are upgrading your Level I machine to Level II, be sure to keep the Level I BASIC ROM. If you already have a Level II machine or your Level I ROM has two chips, you will have to acquire the single-chip ROM. My dealer quoted a price of $29.95 for the chip.

If you are familiar with digital logic, the following paragraphs will help you understand the logic behind the modification. I also highly recommend that you get a copy of the *TRS-80 Technical Reference Handbook* (Radio Shack Catalog No. 26-2104). It's well worth the $10 for anyone interested in the hardware side of microcomputing and do-it-yourself maintenance.

Theory

Most of the ROM and RAM addresses and peripheral devices in the TRS-80 are selected by memory mapping. That is, each device is assigned addresses in the 65K range of possible memory addresses for the Z-80 CPU. For example, the Level I ROM occupies address spaces 0000H through 0FFFH. Level II ROM occupies 0000H through 2FFFH. RAM is allocated to 4000H through FFFFH.

Addresses 3000H through 3FFFH are used to map TRS-80 peripherals, including the display (3C00H through 3FFFH) and keyboard (3800H through 38FFH). There's a lot of unused address space in this range and, someday, I'm going to figure out a use for it.

In order to select the appropriate 4K segments of address space, the TRS-80 uses a three line to eight line decoder (Z21 in Fig. 1) to translate the most significant hex digit of the address (bits A12–A15). The 3 translation is used on the CPU board for peripheral device selection. The remaining seven translations are fed to a 16-pin DIP socket (X3), which is used for memory mapping.

Plugged into socket X3 is a programmable DIP shunt. The DIP shunt is nothing but shorting bars between pins 1 and 16, 2 and 15, etc. To program the DIP shunt, you merely break the appropriate shorting bars, resulting in an open circuit.

The "outputs" of the DIP shunt are ROM A*, ROM B* and RAM* (the * is Radio Shack's way of indicating a logical NOT or active-low condition). ROM A* is used as a chip select line to ROM socket Z33 and ROM B* as a chip select line to ROM socket Z34.

RAM* is used to select the onboard RAM sockets and covers the address range 4000H through 7FFFH. Memory addresses 8000H and above are off-board (i.e., in the expansion

TRS-80* — CONDENSE

The Ultimate in BASIC Compression Utilities

** Release 1.3 Now Available **

- Write BASIC programs using single statement lines for ease of maintenance.
- Write BASIC programs with unlimited remarks and comments to improve program readability and documentation....

— AND STILL GET —

OPTIMUM USE OF MEMORY — FASTER PROGRAM EXECUTION

- Compresses programs up to 70% of original size
- Improves execution time by as much as 30%
- Creates multiple-statement program lines
- Blank compression
- Remark and comment deletion
- Renumbers GOTO, GOSUB, THEN, ELSE, and RESUME statements which reference deleted line numbers
- PLUS THESE NEW USER REQUESTED OPTIONS:
  - Retention of low numbered remark statements
  - Checkpoint / Restart Facilities
  - Phase 1 work file

Model I $21.95 (Diskette)

Model II $24.95 (Diskette)

INTERNATIONAL SOFTWARE ASSOCIATES
P.O. Box 14805
Omaha, Ne. 68124
187

Tandy Corporation*

MICRO-SPEED

TRS-80 SPEED UP MODIFICATION

THE BEST ON THE MARKET!

Simutek's Micro-Speed increases computer operating speed by 50%

1) Speeds up your TRS-80 from 1.77MZ to a hefty 2.66 MZ!
2) Shuts down during disk or cassette I/O to end lost programs, then turns back on automatically when disk or cassette I/0 is finished. (Provided, of course, it was on to begin with)
3) Connects to keyboard L.E.D. and blinks when unit is operating. Stops blinking when turned off!
4) Comes with illustrated instructions. (Some soldering required)
5) Average person can install in 10-20 minutes!
6) All work is done in hardware! Absolutely no software drivers needed!
7) Operates with any TRS-80, (except Model II) works with TRSDOS, NEWDOS or any other operating system or software. Works with RS-232 and telephone modems!
8) Comes completely assembled and ready to install. All wires are stripped, switch installed!

There are no hidden extra costs to you!

9) May be turned on or off at any time! Even during program execution!
10) Money back guarantee!!

order #2000 Micro-Speed Mod. **29.95**

We accept Visa · Master Charge · Money Order · Check (C.O.D. $3.00 extra) Send orders to:
Simutek, P.O. Box 13687-Z, Tucson, AZ 85732 19

Name _____

Address _____

City _____ State _____ Zip _____

Phone orders welcome 24 hours! (800) 528-1149 Simutek offers a number of other fine products especially for TRS 80's! Send for "FREE" catalog. TRS 80 is a TM of Radio Shack, A Tandy Corp. Arizona residents add 4% sales tax

NO CHARGE FOR SHIPPING

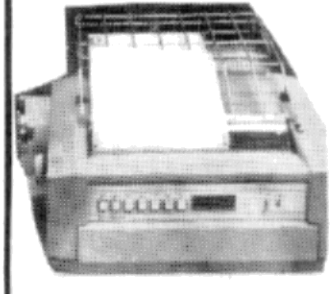
RONDURE COMPANY

191

2522 BUTLER ST
DALLAS, TEXAS 75235
214-630-4621

the computer room

SPECIAL
Printer for your Microcomputer



GE TERMINET 300 PRINTER

Pin feed—9" paper

- 80 Print positions
- Receive only
- ASCII code
- RS-232 interface
- 30 CPS
- Upper & lowercase
- Shipping wt 75#

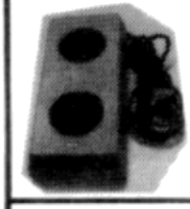
Shipping containers $15.00.

(used)

(good working condition)

Will run on serial RS232 port of most micros including TRS-80. **$450.00**

WE HAVE FLAT-PACK ACOUSTIC



Modem pickup

$19.50

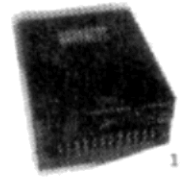
USED FANS

Muffin — 8.00
Sprite — 4.00

NEW POWER SUPPLY

$25.00

5V at 3 Amp
12V at 6 Amp
-12V at 3 Amp



USED POWER SUPPLY

$15.00

5V at 12 Amp
16V at 6 Amp
6V at 2 Amp

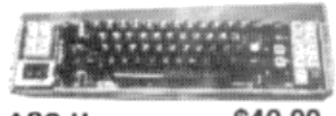
NEW POWER SUPPLY

(AC-DC Brand)

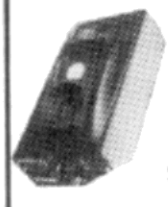
Model 1-22V @ 1.9a $20.00
Model 2-15V @ 2.4a $30.00

MICRO SWITCH KEYBOARD

USED BUT LOOKS VERY NICE



ASC II **$40.00** (With Print)



USED OMNITEK

MODEM
ORIGINATE
ONLY
TESTED

$90.00 Sale

ORDERING INFORMATION:
We ship the same day we receive a certified check or money order. Texas residents add 5% sales tax. Write for our CATALOG of many parts, terminals, printers, etc. All items subject to availability. Your money returned if we are out of stock. Mail order hours 9-4 Monday thru Thursday (closed Friday).

SHIPPING INFORMATION:
Modems: $3.00 each; Key Boards $4.00, Power Supply $7.00.
Large Items & Parts: Specify Freight or Air Freight Collect.
Foreign Orders: Add appropriate freight or postage.
We now take Master Charge and Visa orders. Specify full number, bank number and expiration date.

interface). Note in Fig. 1 that A15* AND RAS* (row address select) are used to enable all on-board memory translations.

ROM A* (pins 7 and 8 of X3) is used in a Level II machine to select the outrigger circuit board containing the Level II BASIC ROM chips.

If you follow the wiring of the DIP shunt in Fig. 1, you will see that ROM A* is low (active) for translations of 0, 1 or 2. ROM B* is also low in this configuration, but not necessary for Level II ROM selection.

By now, you can probably see the necessary changes taking form. If the single-chip Level I BASIC ROM is in socket Z34, and we can somehow change the programming of the DIP shunt 0 translation to ROM A* or ROM B* at will, we have all the necessary ingredients.

We don't have to worry about the 1 and 2 translations, since they will always select ROM A* (i.e., Level II ROM), and the Level I ROM will never reference addresses in that range.

Modification

Assuming that you have the single-chip BASIC I ROM, the only other parts you will need are a SPDT switch, a 16-pin DIP header (Radio Shack Catalog No. 276-1980) and three pieces of flexible hook-up wire six to eight inches long. Any SPDT switch, such as a good subminiature toggle switch (Radio Shack Catalog No. 275-613), will do. I used a three-strand ribbon cable for the wire.

Fig. 2 illustrates the wiring of the DIP header and switch. The procedure is as follows:

1. Cut three six to eight-inch pieces of hook-up wire or ribbon cable. Trim ¼ inch of insulation from each end and tin the leads.
2. Connect one lead (center of the ribbon cable) between pin 10 of the DIP header and the center pole of the SPDT switch. Solder both ends.
3. Connect one lead between pin 6 of the DIP header and one of the remaining SPDT switch poles. This is the Level I position. Solder both ends.
4. Connect the last lead between pin 9 of the DIP header and the remaining SPDT switch pole. This is the Level II position. Solder the switch end only.
5. Connect a wire jumper between pins 8 and 9 of the DIP header. Solder both ends.
6. Connect a wire jumper between pins 7 and 11 of the DIP header. Solder both ends.
7. Connect a wire jumper between pins 2 and 15 of the DIP header. Solder both ends.
8. If you have a 16K machine (or larger), connect wire jumpers between pins 3 and 14, 4 and 13, and 5 and 12. Solder all ends.

This completes wiring of the DIP header and switch assembly. Now, on to the installation:

9. Disconnect all cables to the keyboard unit and lay it upside-down on a nonmarring surface. Remove the six Phillips screws in the bottom.
10. Holding the case top and bottom together, turn it over and carefully lift off the top. Be careful of the power-on LED.
11. Carefully raise the keyboard assembly and fold it back. Do not strain the interconnecting

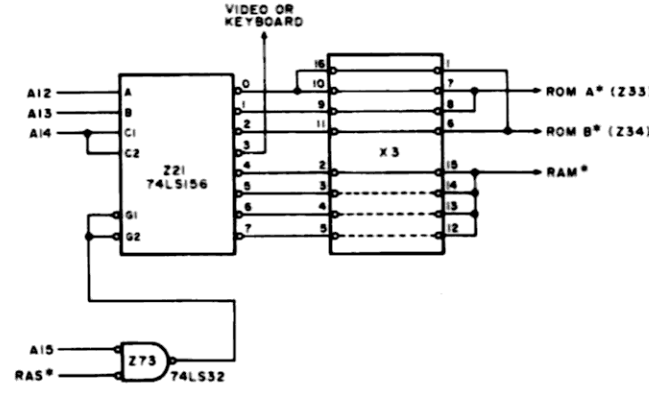


Fig. 1. Memory Mapping in the TRS-80 Level II.

4 MHz TRS-80*

MODEL I UPGRADE

PLUG COMPATIBLE PROCESSOR

- **DOUBLES PROCESSING POWER** of Level I and II by increasing speed 2.25 times to equal Model II.
- **RELIABLE 4 MHz OPERATION** USES Z-80A - not a mere clock speed up.
- **EASY TO INSTALL** - plugs into expansion port.
- **COST EFFECTIVE** - costs less than 5% of typical system price.

DIMENSIONS: 13" L 3¼" H 2¾" W
COLOR: Grey
POWER REQUIREMENTS: 120 VAC

Send check for: **$200.00**

MicroCompatible Inc. ✓458
P. O. Box 107
Scaly Mt., N. C. 28775

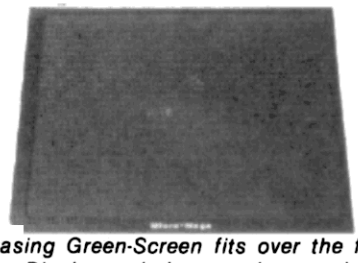
includes shipping and handling

(704) 526-2782

*Registered Trademark of Tandy Corp.

for the TRS-80 from Micro-Mega

The Original GREEN-SCREEN



The eye-pleasing Green-Screen fits over the front of your TRS-80 Video Display and gives you improved contrast with reduced glare. You get bright luminous green characters and graphics like those featured by more expensive CRT units.

Don't confuse the Original Green-Screen with a piece of thin film stuck to the face of your video tube, such as that advertised by others. The Original Green-Screen is mounted in a full frame perfectly matched to the color and texture of the TRS-80 Video Display. It is attached with adhesive strips which do not mar your unit in any way.

The full frame design of the Original Green-Screen "squares off" the face of your video display and greatly improves the overall appearance of your system.

THE GREEN-SCREEN.....*$13.95*
Add $1.00 for postage and handling.

Terms: Check or money order, no CODs or credit cards, please. Add amount shown for postage and handling to price of the item. All items shipped within 48 hours by first class or priority mail. Virginia residents, add 4% sales tax.

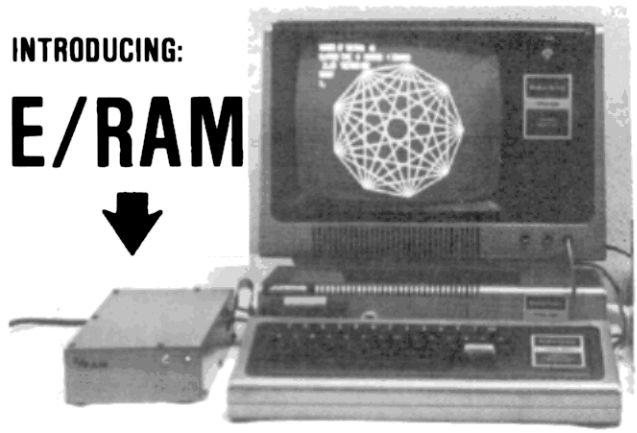
✓29

Micro-Mega · P.O. Box 6265 · Arlington, Va 22206

HI-RESOLUTION GRAPHICS FOR TRS-80*

INTRODUCING:

E/RAM



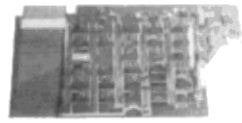
E/RAM Graphics is a unique hardware/software package, which will integrate high-speed, high resolution graphics into any Level II TRS-80 system. E/RAM hardware is a fully plug-compatible box, which installs in minutes, and requires absolutely **no** modifications to the TRS-80 system. E/RAM software is a compact, relocatable set of utilities which provides the user with easily accessible graphics functions. For instance: the user pokes the end point coordinates of a line into certain locations, does a USR call, and an optimized dot-raster line is automatically drawn on the screen at very high speed (less than 10 milli-seconds for a medium length line).

E/RAM does not require the purchase of an additional monitor CRT. The high-resolution graphics video is syncronized with the TRS-80 video and appears on the screen with the normal TRS-80 display. Alphanumerics, TRS-80 graphics, and E/RAM high-resolution graphics may be displayed simultaneously or individually.

E/RAM hardware contains its own 6144 byte video memory, which provides a true 256 x 192 matrix of **independent** graphic elements. (E/RAM is **NOT** a programmable character generator type graphics system. Character generator systems have serious limitations in full screen graphics applications.)

E/RAM will operate with or without an expansion interface, and with any standard memory configuration (4k through 48k).

E/RAM is **fast**. "E/RAM" is an acronym for **E**xtended **R**andom **A**ccess **M**emory, a very short description of the Patent-Pending method of I/O employed by this device, which gives it memory-mapped speed without interfering with the memory space used by the TRS-80.



The installation of E/RAM will not affect normal operation of the TRS-80. High resolution ON/OFF is under program or manual control (a switch is provided). An expansion card edge connector is provided so that other peripherals may be used on the TRS-80 bus.

E/RAM software package is compact (less than 1000 bytes), fast, easy to use, and very flexible. A relocating loader is provided. The user can delete unneeded routines if more memory space is required. Lines can be drawn as fast as 13 per second using BASIC USR calls, and as fast as 200 per second using assembly language programs.

Routines usable through USR of BASIC, and of course an assembler CALL are:

INIT	-	Sets up display
PLOT	-	Plots a point
READ	-	Reads a point from the screen
BLACK	-	Sets drawing mode to black (off)
WHITE	-	Sets drawing mode to on
CLEAR	-	Clears the high-resolution graphics screen
LINE	-	Draws a line

As an example, after the utilities package is loaded and you desire to draw a line, the following sequence of BASIC instructions could be executed:

U=USR(0)	Return the communications area
POKE U+1,X0	Provide the beginning X coordinate
POKE U+3,Y0	Provide the beginning Y coordinate
POKE U+5,X1	Provide the ending X coordinate
POKE U+7,Y1	Provide the ending Y coordinate
V=USR(4)	Draw the line (Current speed is approximately 13 vectors/second)

The complete E/RAM package is available for only $349.95, and includes case, power supply, cables, software cassette, and complete documentation.

To order, or for further details, write or call:

VERN STREET PRODUCTS ✓432
114 West Taft
Sapulpa, Oklahoma 74066
Phone: (918) 224-5347

We handle a full line of Radio Shack products

Dealer inquiries are invited.
Terms: COD Welcome, check, money order, Master Charge, or Visa
Delivery: Stock to 60 days.
E/RAM was designed, and is manufactured by KEYLINE COMPUTER PRODUCTS, INC. 13 East 6th Street, M/C 200, Tulsa, Oklahoma 74119.

*TRS-80 is a registered trademark of Radio Shack, a Tandy Corporation.

ribbon cable.

12. Remove the five rubber separators. Remember where they were.
13. Carefully remove the CPU logic assembly together with the keyboard assembly and turn them over to the component side.
14. Remove the DIP shunt at X3 (near the expansion interface edge connections) and plug the DIP header/switch assembly in its place. Make sure it is plugged in correctly.
15. If the ribbon cable from the Level II board is plugged into socket Z34, carefully remove the plug and insert it into socket Z33.
16. Make sure the single-chip BASIC ROM is positioned correctly and carefully insert it into socket Z34.
17. At this point, you'll have to decide where and how to mount the SPDT switch. I drilled a hole near the center of the rear apron of the case. Make sure the wire will reach the switch position when the PC board is reinstalled in the case.
18. You may want to check out the computer before replacing it in the case. See Checkout and Operation.
19. Reassemble the unit by reversing the above procedure, installing the switch as you go. Be sure that the five rubber separators are on the posts under the keyboard assembly. If you mixed up the screws, the shortest ones go in the holes near the front of the keyboard, the middle-sized screws in the two center holes and the long ones in back.

Checkout and Operation

Connect the power supply and display cables. Put the switch in the Level II position and turn the power on. The video should display:

```
MEMORY SIZE?_
```

If the display is:

```
READY
>
```

the switch is in the Level I position.

If the display is garbage, turn the power off, wait a few seconds and turn the power on again. If the display is still garbage or clear, turn the power off and recheck all your wiring. Make sure the DIP header, BASIC I ROM chip and ribbon cable connector from the Level II board are properly oriented in their sockets.

Now, turn the power off and switch to the other Level position. Wait a few seconds and turn the power on. The display should now be the alternate initial message. If Level II worked but Level I does not, the problem is likely in the Level I ROM chip, its orientation or the wire from the Level I position on the switch.

You'll have to turn power off each time you change levels. If you just throw the switch, the CPU usually goes out to lunch. The manual reset won't recover.

The reason for pausing each time you turn the power off and on is to allow time for the power-on reset capacitor to discharge. ■

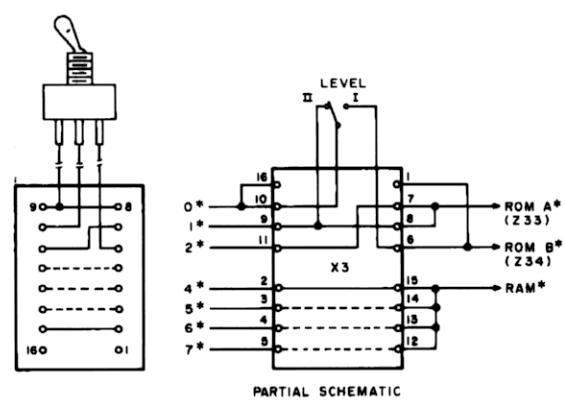


Fig. 2. Switch/Header Assembly.

Genetic counsel from your TRS-80.

Genotype

Albert Rauber M. D.
Department of Pediatrics
69 Butler St., S. E.
Atlanta, GA 30303

Prospective parents faced with genetic problems must make their own decisions. Only they can balance the risks and benefits of pregnancy, abortion, contraception, sterilization or adoption.

Their physician will often tell them what the odds are that a child would have a particular birth defect or would carry the trait. But these statistics come from large samples, and can be misleading. No woman will have enough pregnancies in her lifetime to demonstrate the odds.

This program gives the patient a more realistic view of probability as applied to human genetics. It uses four examples:

- The general inheritance of an autosomal recessive gene that requires a contribution from each parent. ("Autosomal" refers to chromosomes other than those that determine sex.)
- Hemophilia, which illustrates the workings of a gene carried on the special chromosomes that determine sex.
- Sickle-cell disease, which differs from other autosomal recessive genes in that we can test for it even when it causes no disease.
- The four major blood types, which are combinations of two genes, each of which expresses itself as a single gene and neither of which may be present.

The program is written on a TRS-80 Level II in BASIC and runs on 16K. It comprises 8357 bytes.

The Program

Statements 300-360 select topics from a menu. 1000-1200 pick the genotypes of each parent for the recessive inheritance example.

1210 sends the program to 20000, a subroutine to print and label a conventional pedigree tree. A second branching at 20040 and 20070 goes to subroutines at 30000, which fill in the male and female symbols to indicate affected or carrier states.

20302 defines the number of children to be born. 20305 combines the parents genotypes, and branches to appropriate lines. 20307-20329 calculate the probabilities using the random number generator. 20332 counts the children, and 20329 determines sex.

20334 branches to a subroutine that defines the x,y coordinates for printing the next child on the family tree. Line 20440 directs the program to graphic subroutines, which print the symbol appropriate to sex and genotype (30018-30450). Line 20470 lets you repeat the experiment or return to the menu.

The other examples use the same graphics subroutines but differ in their calculations.

The hemophilia (x linked recessive) program considers only two genotypes (G) and the two sexes (R), which are determined by serial RND (X) statements to produce four different outcomes in lines 2130-2160.

The blood type example, beginning at line 3000, uses string functions, since blood type names are actually concatenations of the gene names (A,B,O). Although sex has nothing to do with the inheritance, I've kept the symbols and terms to enhance realism.

Line 3010 summarizes a basic information review. Lines 3110-3170 ask for the paired genes of each parent. Lines 3250-3210 randomly select which gene each parent will furnish to the conceptus (C$) in line 3320.

Lines 3330 and invoked subroutines print and label the family tree, with genotypes for the blood groups.

The sickle-cell routine at line 4000 follows the same basic pattern as the blood group routine, but with different text.

Repeated runs compress time and let you simulate many families of varying sizes, giving a more accurate feel for the odds and stakes involved.

Biology teachers can modify this program for classroom use by removing the medical references and using sweet pea flower colors, guinea pig coat markings or other characteristics. ■

Program Listing 1.

```
5 CLS
10 PRINT"COPYRIGHT 1979 ALL RIGHTS RESERVED. ALBERT RAU
     BER DECATUR, GA.":INPUTZ$
100 CLEAR
110 CLS
200 PRINT CHR$(23)
210 PRINT@468,"GENOTYPE":FORI=1TO1000:NEXTI:CLEAR
212 CLS
300 PRINT"SELECT THE MODE OF INHERITANCE FROM THIS LIST
     .":PRINT
310 PRINT"ENTER THE NUMBER OF YOUR CHOICE."
320 PRINT TAB(10) "(1) AUTOSOMAL RECESSIVE"
330 PRINT TAB(10) "(2) X LINKED RECESSIVE (HEMOPHILIA)"
340 PRINT TAB(10) "(3) AUTOSOMAL DOMINANT (BLOOD TYPES)
     "
```

Program continued

```
341 PRINT TAB(10)"(4) SICKLE CELL ANEMIA"
350 PRINT@896,"":INPUTZ
360 CLS:ON Z GOTO 1000,2000,3000,4000
370 STOP
999 CLS
1000 PRINT"          THIS PROGRAM ILLUSTRATES THE PROPGA
     TION OF AN AUTOSOMAL          RECESSIVE TRAIT"
1001 PRINT@896,"PRESS ENTER":INPUT Z$
1002 CLEAR
1004 RANDOM
1100 CLS:PRINT"CHOOSE A GENETIC TYPE FOR EACH PARENT":P
     RINT:PRINT
1110 PRINT"THE FATHER SHALL BE:"
1111 PRINT
1120 PRINT TAB(10)"(1) NORMAL"
1130 PRINT TAB(10)"(2) AFFECTED"
1140 PRINT TAB(10)"(3) CARRIER"
1141 PRINT
1150 INPUT"TYPE THE NUMBER OF YOUR CHOICE";E
1151 PRINT
1160 PRINT"THE MOTHER SHALL BE:":PRINT
1170 PRINT TAB(10)"(1) NORMAL"
1180 PRINT TAB(10)"(2) AFFECTED"
1190 PRINT TAB(10)"(3) CARRIER":PRINT
1200 PRINT:INPUT"TYPE THE NUMBER OF YOUR CHOICE";F
1210 GOTO 20000
2000 ' * X LINKED RECESSIVE TRAIT
2001 CLEAR
2002 CLS
2003 RANDOM
2010 PRINT"   THIS PROGRAM ILLUSTRATES THE PROPOGATION
     OF AN  X LINKED     RECESSIVE TRAIT SUCH AS HEMOPH
     ILIA."
2012 PRINT@769,"PRESS ENTER WHEN READY":INPUTZ$
2014 CLS
2020 PRINT@18,"NORMAL MALE"
2030 PRINT@34,"CARRIER FEMALE"
2034 X=44:Y=6
2040 GOSUB30018
2050 PRINT@343,"X Y"
2054 X=78:Y=6
2060 GOSUB 30174
2070 PRINT@359,"X X'"
2080 GOSUB 30400
2090 X=0:Y=0
2100 PRINT@768,"HOW MANY CHILDREN ( 1 TO 5)";
2104 INPUT K
2108 FOR Q = 1 TO K
2112   C=C+1
2114 ON C GOSUB 30510,30520,30530,30540,30550
2120   R=RND(2):G=RND(2)
2130   IF R=1 AND G=2 THEN A=A+1:P=1:GOTO2170
2140  IF R=1 AND G=1 THEN P=2:GOTO2170
2150  IF R=2 AND G=2 THEN B=B+1:P=3:GOTO2170
2160  IF R=2 AND G=1 THEN P=4:GOTO2170
2170 ON P GOSUB 30300,30018,30174,30110
2176 NEXT Q
2180 PRINT@768,"PRESS ENTER                          ":IN
     PUTB$
2190 PRINT@768,"THERE ARE ";A;" AFFECTED BOYS AND ";B;"
      CARRIER GIRLS."
2200 PRINT@832,"TYPE 1 TO REPEAT, 2 TO RETURN TO MENU,
     3 TO END":INPUT S
2210 ON S GOTO 2000,212,30000
2999 STOP
3000 REM -----BLOOD TYPES ------
3002 CLS:CLEAR(1000)
3010 PRINT"THIS PROGRAM ILLSTRATES THE ACTION OF DOMINA
     NT GENES."
3020 PRINT"BLOOD TYPES A,B,AND O ARE USED AS EXAMPLES."
     :PRINT
3030 PRINT"THE GENE A PRODUCES ANTIGEN A; B GENE PRODUC
     ES ANTIGEN B AND"
3040 PRINT"EACH GENE EXPRESSES ITSELF INDEPENDENTLY OF
     THE OTHER."
3050 PRINT"O REPRESENTS AN ABSENT GENE; IF BOTH A AND B
      ARE ABSENT"
3060 PRINT"NO ANTIGEN IS PRESENT AND TYPE O EXISTS.":PR
     INT
3070 PRINT"A TYPE A PERSON MAY RESULT FROM GENOTYPE 'AA
     ' OR 'AO' BUT"
3080 PRINT"THE 'AO' INDIVIDUAL WILL TRANSMIT THE 'A' GE
     NE TO PROGENY IN"
3090 PRINT"ONLY ONE HALF OF THE INSTANCES."
3100 PRINT@896,"PRESS ENTER WHEN READY";:INPUTZ$
3108 CLS
3110 PRINT"CHOOSE BLOOD TYPES FOR THE PARENTS."
3112 PRINT
3120 C=C+1: IF C=1 THEN P$="FATHER" ELSE P$="MOTHER":PR
     INT
3130 PRINT"TYPE A COMBINATION OF 2 GENES FOR THE ";P$;"
      SUCH AS"
3140 PRINT"AA, AO, BB, BO, AB, OO":PRINT
3150 J=J+1: IF J=2 GOTO 3170
3160 INPUT GF$: GOTO 3120
3170 INPUT GM$
3172 CLS
3175 PRINT@151,GF$
3180 PRINT@168,GM$
3190 X=44:Y=5:GOSUB 30018
```

Program continues

```
3200 X=80:Y=5: GOSUB 30110
3220 GOSUB 30410
3230 X=0:Y=0
3240 PRINT@ 768,"HOW MANY CHILDREN ( 1 TO 5 )";:INPUT O
3250 FOR Q= 1 TO O
3260   GF=RND(2):ONGF GOTO 3270,3280
3270   F$=MID$(GF$,1,1):GOTO 3290
3280   F$=MID$(GF$,2,1)
3290   GM =RND(2):ON GM GOTO 3300,3310
3300   M$=MID$(GM$, 1,1):GOTO3320
3310   M$=MID$(GM$, 2,1)
3320   C$= F$+M$
3330   K=K+1:ON K GOTO 3332,3334,3336,3338,3340
3332   PRINT@719,C$:GOTO3350
3334   PRINT@727,C$:GOTO3350
3336   PRINT@735,C$:GOTO3350
3338   PRINT@743,C$:GOTO3350
3340   PRINT@751,C$:GOTO3350
3350   ON K GOSUB 30510,30520,30530,30540,30550
3360   S=RND(2):ON S GOSUB 30018,30110
3370 NEXTQ
3380 PRINT@896,"PRESS ENTER";:INPUTZ$
3390 PRINT@896,"TYPE 1 TO REPEAT, 2 TO RETURN TO MENU,
     3 TO END"
3400 INPUTZ:ONZ GOTO 3000,212,30000
3999 STOP
4000 REM ------SICKLE CELL-------
4002 CLEAR
4004 CLS
4006 RANDOM
4010 PRINT"THIS PROGRAM SHOWS HOW THE GENES FOR SICKLE
     HEMOGLOBIN"
4020 PRINT"ARE PASSED ON TO PROGENY.  A PERSON RECEIVIN
     G A SICKLE"
4030 PRINT"GENE (S) FROM EACH PARENT (SS) MAKES ABNORMA
     L SICKLE"
4040 PRINT"HEMOGLOBIN AND BECOMES SICK. A PERSON WHO RE
     CEIVES ONLY"
4050 PRINT"ONE (S) GENE BUT ALSO HAS AN (A) GENE MAKES
     ENOUGH SICKLE"
4060 PRINT"HEMOGLOBIN TO DETECT BUT DOES NOT BECOME ILL
     . HOWEVER,"
4070 PRINT"THE (S) GENE CAN BE PASSED ON TO THE CHILDRE
     N WHO MAY"
4080 PRINT"OR MAY NOT GET SICKLE CELL ANEMIA DEPENDING
     ON WHAT GENE"
4090 PRINT"THEY RECEIVE FROM THE OTHER PARENT. PERSONS
     WITH TWO (A) GENES"
4100 PRINT"(AA) MAKE ONLY NORMAL ADULT HEMOGLOBIN.":PRI
     NT
4110 PRINT"THERE ARE OTHER COMBINATIONS OF OTHER KINDS
     OF HEMOGLOBIN"
4120 PRINT"WHICH MAY CAUSE SICKNESS.":PRINT@896,"PRESS
     ENTER":INPUTZ$
4200 CLS:PRINT"CHOOSE A GENETIC TYPE FOR EACH PARENT"::
     PRINT
4208 P$="FATHER"
4210 PRINT"THE " ;P$;" SHALL:"
4220 PRINT TAB(10)"BE NORMAL  (AA)"
4230 PRINT TAB(10)"HAVE SICKLE CELL ANEMIA  (SS)"
4240 PRINT TAB(10)"HAVE SICKLE CELL TRAIT   (SA)"
4244 IFGF$="SS" THEN GF=3
4250 PRINT"TYPE THE LETTERS REPRESENTING THE GENE TYPE
     OF YOUR CHOICE(";
4251 C=C+1:IFC=2GOTO4256
4252 INPUT GF$:PRINT@447,")"
4254 PRINT:P$="MOTHER":GOTO 4210
4256 INPUT GM$:PRINT@895,")"
4258 CLS
4259 PRINT@343,GF$
4260 IFGF$="AA" THEN GF=1:GOTO4265
4262 IFGF$="SA"ORGF$="AS" THEN GF=2:GOTO4265
4264 IF GF$="SS" THEN GF=3
4265 X=44:Y=6:ON GF GOSUB 30018,30360,30300:GOSUB30410
4266 GOSUB 30410
4268 PRINT@359,GM$
4270 IF GM$="AA" THEN GM=1:GOTO4276
4272 IF GM$="SA"OR GM$="AS" THEN GM=2:GOTO4276
4274 IF GM$="SS" THEN GM=3
4276 X=78:Y=6:ONGM GOSUB 30110,30174,30235
4280 GOSUB 30410
4300 PRINT@768,"HOW MANY CHILDREN (1 TO 5)";:INPUT O
4350 FOR Q=1TO O
4360   F=RND(2):ON F GOTO 4370,4380
4370   F$=MID$(GF$,1,1):GOTO4390
4380   F$=MID$(GF$,2,1)
4390   M=RND(2):ON M GOTO4400,4410
4400   M$=MID$(GM$,1,1):GOTO4420
4410   M$=MID$(GM$,2,1)
4420   C$=F$+M$
4430    K=K+1:ON K GOTO 4432,4434,4436,4438,4440
4432    PRINT@719,C$:GOTO4450
4434    PRINT@727,C$:GOTO4450
4436    PRINT@735,C$:GOTO4450
4438    PRINT@743,C$:GOTO4450
4440    PRINT@751,C$:GOTO4450
4450    ON K GOSUB 30510,30520,30530,30540,30550
4460    IF C$="AA" THEN V = 0
```

Program continued

```
4490 PRINT@960,"TYPE 1 TO REPEAT,2 TO RETURN TO MENU AN
     D 3 TO END":INPUTZ:ONZGOTO4000,212,30000
4462    IF C$="SA" OR C$="AS" THEN V=2
4464    IF C$="SS" THEN V=4
4466    S=RND(2): ONS+V GOSUB30018,30110,30360,30174,30
     300,30235
4468 IFS+V=1 THEN A=A+1
4470 IFS+V=2 THEN B=B+1
4472 IFS+V=3 THEN D=D+1
4474 IFS+V=4 THEN E=E+1
4476 IFS+V=5 THEN H=H+1
4478 IFS+V=6 THEN J=J+1
4479 NEXT Q
4480 PRINT@768,"THERE ARE";H;" BOYS AND ";J;" GIRLS WHO
      DEVELOP SICKLE CELL ANEMIA. tHERE ARE";D;" BOYS A
     ND";E;" GIRLS WHO CARRY THE TRAIT WITHOUT BEINGSIC
     K AND ";A;" BOYS AND ";B;" GIRLS WHO ARE NORMAL.
     (PRESS ENTER)":INPUTZ$
4999 STOP
5000 CLS
5002 FOR I=1TO1000:NEXTI
5010 END
5256 INPUT F
20000 ' * PRINTPEDIGREE
20005 CLS
20010 PRINT@21," MALE"
20020 PRINT@37," FEMALE"
20030 X=44:Y=6
20040 ON E GOSUB 30018,30300,30360
20050 ON E GOSUB20052,20053,20054:GOTO20060
20052 PRINT@340,"NORMAL":RETURN
20053 PRINT@339,"AFFECTED":RETURN
20054 PRINT@340,"CARRIER":RETURN
20060 X=80:Y=6
20070 ON F GOSUB 30110,30230,30174
20080 ON F GOSUB20082,20083,20084:GOTO20290
20082 PRINT@357,"NORMAL":RETURN
20083 PRINT@357,"AFFECTED":RETURN
20084 PRINT@357,"CARRIER":RETURN
20290 GOSUB 30410
20300 X=0:Y=0
20302 PRINT@768,"HOW MANY CHILDREN? (1 TO 5)":INPUT O
20304 FOR Q = 1 TO O
20305 ON E+F GOTO 20306,20307,20308,20309,20313,20315
20306 CLS:PRINT"ERROR - START OVER":FORI=1TO250:NEXTI:G
     OTO5
20307 T=2:B=B+1:GOTO 20329
20308 T=4:D=D+1:GOTO 20329
20309 IF E<>FGOTO20311
20310 T=0:A=A+1:GOTO 20329
20311 H = RND(2)
20312 ON H GOTO 20320,20321
20313 H = RND(2)
20314 ON H GOTO 20319,20320
20315 H= RND(3)
20316 ON H GOTO 20319,20320,20321
20319 T=0:A=A+1:GOTO20329
20320 T=4:D=D+1:GOTO20329
20321 T=2:B=B+1:GOTO 20329
20322 R=RND(2)
20324 G=RND(2)
20326 IF R=1 AND G=1 THEN T=0:A=A+1
20327 IF R=2 AND G=2 THEN T=2:B=B+1
20328 IF R<>G THEN T=4:D=D+1
20329 S=RND(2): P=T+S
20332 C=C+1
20334 ON C GOSUB 30510,30520,30530,30540,30550
20440 ON P GOSUB 30300,30230,30018,30110,30360,30174
20442 X=0:Y=0
20445 NEXT Q
20460 PRINT@768,"PRESS ENTER                          "
     :INPUTB$
20470 PRINT@768,"THERE ARE ";A;" AFFECTED CHILDREN AND
     ";O-A;" UNAFFECTED CHILDREN OF  WHOM ";D;" ARE CAR
     RIERS OF THE TRAIT."
20500 PRINT:PRINT:PRINT"TYPE 1 TO REPEAT, 2 TO RETURN T
     O MENU, 3 TO END":INPUT Z
20510 ON Z GOTO 999,212,30000
30000 CLS:PRINTCHR$(23):PRINT@474,"GOODBYE":FORI=1TO100
     0:NEXTI:CLS:FORI=1TO1000:NEXTI:END
30010 REM * PRINT SYMBOLS
30018 REM * NORMAL MALE
30020 Y=Y:FORX=XTOX+7:SET(X,Y):NEXT X
30030 X=X:FORY=YTOY+5:SET(X,Y):NEXTY
30040 Y=Y:FORX=X TO X-8STEP-1:SET(X,Y):NEXT X
30050 X=X:FORY=YTOY-6STEP-1:SET(X,Y):NEXTY
30060 RETURN
30062 STOP
30100 '* NORMAL FEMALE
30110 Y=Y:FORX=XTOX+4:SET(X,Y):NEXTX
30120 X=X+1:FORY=Y+1TOY+4:SET(X,Y):NEXTY
30140 Y=Y:FORX=X-2TOX-4STEP-1:SET(X,Y):NEXTX
30150 X=X-1:FORY=Y-1TOY-4STEP-1:SET(X,Y):NEXTY
30160 RETURN
30162 STOP
30170 '* CARRIER FEMALE
30174 GOSUB30110
30180 Y=Y+1:FORX=X+1TOX+6:SET(X,Y):NEXTX
30190 Y=Y+1:FORX=X-2TOX-6STEP-1:SET(X,Y):NEXTX
30200 Y=Y+1:FORX=X+1TOX+4:SET(X,Y):NEXTX
30210 Y=Y+1:FORX=X-2TOX-3STEP-1:SET(X,Y):NEXTX
30220 Y=Y+1:X=X+2:SET(X,Y)
30222 RETURN
30224 STOP
30230 ' AFFECTED FEMALE
30235 GOSUB 30110
30240 FOR N=1TO5
30245 Y=Y+1:FORX=XTOX+8:SET(X,Y):NEXTX
30250 X=X-9::NEXTN
30255 RETURN
30260 STOP
30300 '*AFFECTED MALE
30310 FOR N=1TO7
30320 Y=Y:FORX=XTOX+8:SET(X,Y):NEXTX
30330 X=X-9:Y=Y+1
30340 NEXT N
30342 RETURN
30350 STOP
30352 STOP
30360 ' CARRIER MALE
30365 GOSUB 30020:U=8
30370 FOR N=1TO6
30375   Y=Y+1:FORX=XTOX+U:SET(X,Y):NEXTX
30380   X=X-(U+1):U=U-1
30385 NEXT N
30390 RETURN
30395 STOP
30400 '* TREE
30410 Y=8:FORX=52TO77:SET(X,Y):NEXTX
30420 FORX=65TO66:FORY=9TO20:SET(X,Y):NEXTY:NEXTX
30430 Y=21:FORX=33TO97:SET(X,Y):NEXTX
30440 FORX=33TO97STEP16:FORY=21TO24:SET(X,Y):NEXTY:NEXT
     X
30450 RETURN
30452 STOP
30500 ' * COODINATES
30510 X=29:Y=24:RETURN
30520 X=46:Y=24:RETURN
30530 X=62:Y=24:RETURN
30540 X=78:Y=24:RETURN
30550 X=94:Y=24:RETURN


DONE
```


✓Reader Service—see page 226

Hear it with an RS-232 Interface Board.

The Serial Clank on the Printer

William O'Brien
11 Dongan Place
New York, NY 10040

LIST_

How many times do you type that word, hit return and watch as the lines zip past your eyes, and prepare to press both the @ and shift keys at the same time, trying to stop the speeding display at the line you think you want?

Either you pass the line or you stop the listing with the line so close to the top, it clears off when you press the break key. If you're at all like me, your video display blushes at the string of epithets that follows.

Neon signs go off in your mind: *hard copy, printed output!*

The Serial Interface

Now you find yourself thumbing through your favorite microcomputing magazine and you see some ads for a GE terminet or a Teletype 33 or your friend Hubert is ditching his Model 99 Geodeisic impact printer. Upper and lowercase, plain paper. Not bad. The prices are only about $200 to $500. Great! But it needs a serial interface. Well, that kills that, right?

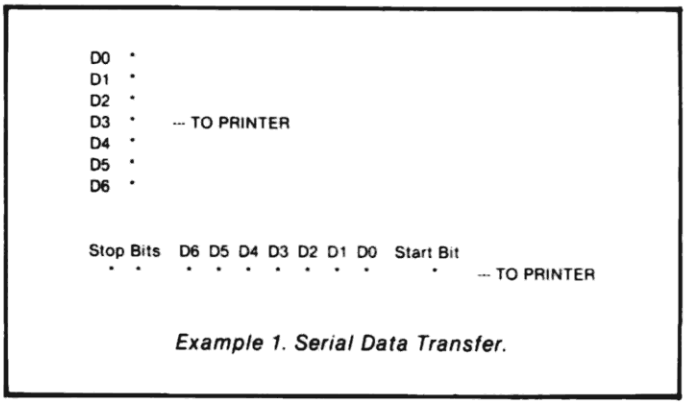
Maybe not. You don't have to settle for a second-hand printer just because it says "designed for the TRS-80" or is made by Radio Shack.

If you think back a moment to the last time you drooled over items for your machine in a computer catalog, you might remember seeing an ad for something you don't see Radio Shack sell for its TRS-80 very often: a board full of parts. An RS-232 serial board to be specific.

"Run a modem/serial printer from your TRS-80!" Good, no switching plugs on the parallel port; you can even use that second-hand printer! And it does work. But although serial output for a printer is not that hard to come by from the TRS-80, it's not all that easy either. Let's take a quick look at serial data transmission in general before we get on to Radio Shack's solution.

Normally, data is sent to the parallel port and out to the printer in this fashion:

Data bits are sent simultaneously with the appropriate status signals between the device and the CPU, all coded in binary form with 3.5 volts or more as a 1 and 0.8 volts or less as a 0. Data transfer is relatively straightforward and at a fairly good speed. Sounds like a good idea, right?



Example 1. Serial Data Transfer.

✓Reader Service—see page 226

But suppose we want the receiving device to send some information back to the CPU? That means we need seven-plus lines out and seven-plus lines back in. That isn't so hard to imagine, except if you assume that the internal device is not in the same room as the CPU, or even in the same city, for that matter. Everyone knows that for that type of distance you use a modem, a device that codes binary output into two tones, one representing the 0, the other the 1. But, if parallel transfer mode is used that means we'll need seven-plus phone lines; hardly convenient or economical.

Serial data is transferred as in Example 1.

Obviously, for communication purposes at least, serial transmission format has a large advantage. Only one line is needed for data out and one for data in; a single phone line. As far as devices immediately available, hardwiring is only needed on a maximum of four lines: data out, data in, ground and, if necessary, DATA SEND READY, a status line that signals whether the device is free for data reception or busy.

It appears that forming serial data out of parallel is not that difficult a problem. All we must do is collect the parallel bits, line them up, add the appropriate start, stop and status bits, and re-transmit the data all in a row. Well, almost.

Different Types of Data

It seems that there are two different types of serial data. One is called TTL serial, and it follows much the same pattern as we described above: Parallel data stored and re-transmitted with added extras. However, standard RS-232 devices are a little more choosy about what type of signals they'll accept. Unfortunately, ordinary TTL levels are not quite good enough.

Instead, an RS-232 device requires −3 volts or less to indicate a 1 and +3 volts or more for a 0. This means that in addition to reforming the signals leaving the CPU, the interface must also transform them into the proper voltage levels necessary for the receiving device to understand them.

As if that isn't enough, different devices may operate at different data transmission speeds (called "baud rates"), which normally range from 110 to 9600 depending on the function of the device.

That's where the RS-232-C comes in. This is the board Radio Shack produces for the TRS-80 that not only provides standard EIA serial level signals, but also provides 110, 150, 300, 600, 1200, 2400, 4800 and 9600 baud, which are directly programmable via a series of switches in a miniDIP arrangement. Other baud rates may be obtained by POKEing different values into the BRG (Baud Rate Generator).

Odd or even parity, no parity, as well as the number of stop and data bits, are also programmable by means of the switches. Just have a tiny screwdriver, or better still, a sharp pencil, handy.

If you are installing the board yourself, be mindful that the newer expansion interfaces are not equipped with a 44-contact edge connector which mates to the serial board. This pinout will be installed by the Radio Shack technician when he installs your board for you.

You're welcome to try to avoid the charge for this (I think its $7.50) but you will void all warranties on your E/I.

All right, you've installed your board and connected the cable between the expansion interface and the serial printer of your choice. You turn your machine on, load a program and type in LIST. Hit ENTER.

The RESET button is on the left rear portion of the CPU by the E/I connector, in case you've forgotten in the excitement.

WHATHAPPENED???

What happened is that the RS-232 board is designed to be used with a modem in mind and Radio Shack's little blurb about a printer, while not invalid since the output is genuine RS-232, is not exactly all-inclusive.

You *can't* simply connect the device and LPRINT your heart out. Down deep in the memory of your CPU, at locations 4025,

```
10 POKE 16421,2:POKE 16422,0:
   POKE 16423,127
30 :
40
    FOR X=32512 TO 32584
50 READ Y
60 POKE X,Y
70 NEXT
    X
80 DATA 229,197,245,58,72,127,254,1,40
90 DATA
    32,62,1,50,72,127,211,232,219,233
100 DATA 230,24
    6,4,50,71,127,211,234
110 DATA 219,233,230,7,33
,63,127,6,0,79,9
120 DATA 126,211,233,241,193,225,219,2
    34
130 DATA 203,119,40,250,121,211,235,254
140 DAT
    A 13,32,4,14,10,24,239,201,34,68
150 DATA 85,102,1
    19,170,204,238,0,0
```

Listing 1. The original driver program supplied by Radio Shack. Line 20 changes the values in locations 4025, 4026 and 4027 (hex) that point to the printer driver routine. The loop beginning at line 40 loads the driver routine into 7F00 (32512), the top of 16K memory.

```
    BF00  E5        PUSH HL
    BF01  C5        PUSH
     BC
    BF02  F5        PUSH AF
    BF03  3A48BF
      LD   A,(BF48H)
    BF06  FE01      CP   01H

      BF08  2828      JR   Z,BF2AH
    BF0A  3E01
     A,01H
    BF0C  3248BF    LD   (BF48H),A
BF0F  D3E8      OUT  (0E8H),A
    BF11  DBE9      IN   A
    ,(0E9H)
    BF13  E6F8      AND  0F8H
    BF15  F
    604       OR   04H
    BF17  3247BF    LD   (BF47H),
    ,A
    BF1A  D3EA      OUT  (0EAH),A
    BF1C  DB
       IN   A,(0E9H)
    BF1E  E607      AND  07H

   BF20  213FBF    LD   HL,BF3FH
    BF23  0600      LD
       B,00H
    BF25  4F        LD   C,A
    BF26  0
    9         ADD  HL,BC
    BF27  7E        LD   A,(HL)
    )
    BF28  D3E9      OUT  (0E9H),A
    BF2A  F1
      POP  AF
    BF2B  C1        POP  BC
    BF2C
 E1       POP  HL
    BF2D  DBEA      IN   A,(0EAH)

       BF2F  CB77      BIT  06H,A
    BF31  28FA      J
    R    Z,BF2DH
    BF33  79        LD   A,C
    BF34
    4  D3EB      OUT  (0EBH),A
    BF36  FE0D      CP

    BF38  2004      JR   NZ,BF3EH
    BF3A  0
E0A     LD   C,0AH
    BF3C  18EF      JR   BF2DH

    BF3E  C9        RET
    BF3F  224455    LD   (5544H
    ),HL
    BF42  66        LD   H,(HL)
    BF43  77
    7         LD   (HL),A
    BF44  AA        XOR  D

    45  CCEEBC    CALL Z,0BCEEH
    BF48  010140
LD  BC,4001H
```

Listing 4. Disassembled listing of the memory area affected by the revised program for 32K systems. Note that the 7F references have now been replaced by BF.

```
 E1       POP  HL
    7F2D  DBEA      IN   A,(0EAH)

       7F2F  CB77      BIT  06H,A
    7F31  28FA      J
    R    Z,7F2DH
    7F33  79        LD   A,C
    7F34
    4  D3EB      OUT  (0EBH),A
    7F36  FE0D      CP

    7F38  2004      JR   NZ,7F3EH
    7F3A  0
E0A     LD   C,0AH
    7F3C  18EF      JR   7F2DH

    7F3E  C9        RET
    7F3F  224455    LD   (5544H
    ),HL
    7F42  66        LD   H,(HL)
    7F43  77
    7         LD   (HL),A
    7F44  AA        XOR  D

    45  CCEEBC    CALL Z,0BCEEH
    7F48  010140
LD  BC,4001H
    7F00  E5        PUSH HL
    7F01  C5        PUSH
     BC
    7F02  F5        PUSH AF
    7F03  3A487F
      LD   A,(7F48H)
    7F06  FE01      CP   01H

      7F08  2828      JR   Z,7F2AH
    7F0A  3E01
    A,01H
    7F0C  32487F    LD   (7F48H),A

7F0F  D3E8      OUT  (0E8H),A
    7F11  DBE9      IN   A
    ,(0E9H)
    7F13  E6F8      AND  0F8H
    7F15  F
    604       OR   04H
    7F17  32477F    LD   (7F47H),
    ,A
    7F1A  D3EA      OUT  (0EAH),A
    7F1C  DB
       IN   A,(0E9H)
    7F1E  E607      AND  07H

   7F20  213F7F    LD   HL,7F3FH
    7F23  0600      LD
       B,00H
    7F25  4F        LD   C,A
    7F26  0
    9         ADD  HL,BC
    7F27  7E        LD   A,(HL)
    )
    7F28  D3E9      OUT  (0E9H),A
    7F2A  F1
      POP  AF
    7F2B  C1        POP  BC
    7F2C
```

Listing 3. Disassembled listing of the memory area affected by the 16K version of the BASIC driver routine. Note the four 7F values.

```
10 POKE 16421,2:POKE 16422,0:
   POKE 16423,191
30 :
40
    FOR X=-16640 TO -16568
50 READ Y
60 POKE X,Y
70 NE
    XT X
80 DATA 229,197,245,58,72,191,254,1,40
90 DAT
    TA 32,62,1,50,72,191,211,232,
    219,233
100 DATA 230,
    246,4,50,71,191,211,234
110 DATA 219,233,230,7,
33,63,191,6,0,79,9
120 DATA 126,211,233,241,193,225,219
    ,234
130 DATA 203,119,40,250,121,211,
    235,254
140 D
    ATA 13,32,4,14,10,24,
    239,201,34,68
150 DATA 85,102
    ,119,170,204,238,0,0
```

Listing 2. The revised program to relocate the driver routine to the top of 32K. The actual addressing is done in line 40. This listing incorporates the change in data values mentioned in the article.

4026 and 4027 (hex) are the pointers that remind the CPU that it has a parallel printing port, and when it hears someone say L-anything, the output goes to that port. If no printer is there, everything hangs up. It's easy to surmount the difficulties, but let's start with first things first.

Check Your Slide Switch

Assuming you have installed the board correctly and set the baud rate and parity switches as necessary, make sure that the slide switch is *not* in the position marked TERM as the owner's manual suggests.

The reason for this is that the instructions in the manual are primarily for using your TRS-80 as a terminal. As such, data is transmitted over Pin 2 of the DB 25 connector and received via Pin 3.

This is appropriate since, as a terminal, your TRS-80 will be receiving data and the standard pin for data reception is Pin 3.

However, your printer wants its data *sent* to it on Pin 3. With the selector slide switch in the COMM position, the functions of the two pins are reversed. All right, we have data in the correct place. Don't try the LLIST experiment again just yet, there is more.

Those three addresses that control the printing have to be altered and a new driver routine placed in memory so the CPU will output to the correct place.

The driver program loads in at 7F00H, which is decimal location 32512 (line 40, the first POKE address). This location is right at the top of 16K of memory. You use 32511 (one byte below the loading point of the program) in response to the MEMORY SIZE__ query.

My TELPAR thermal printer reacts to this instruction set correctly. However, I have a 32K system; loading an 18K program under DOS wipes out the driver program. It must, therefore, be moved up to the top of my memory where it will be safe.

To make things easy, it should be moved up a full 16K. This will place it at the top of a 32K machine. This means that it must go from 7F00H to BF00H. (Each increment of the leading hex digit increases the value of the memory address by 4K since it's in the 16^3 column. So, to move it 16K, we will have to increment it four times—8F00, or 20K, 9F00, or 24K, AF00, or 28K, since the highest hexadecimal numeric value in a single position is nine with the letters A-F being used as ten through 15; and finally BF00 or 32K).

You will notice that line 20 of the original program supplied by Radio Shack contains three POKE statements. They change the values in memory locations 4025H, 4026H and 4027H. We can concern ourselves primarily with the last two of these, POKE 16422 and POKE 16423.

POKE 16422 puts the value 0 into that particular memory location (4026H) and POKE 16423 puts the value 127 (into 4027H). These two memory POKEs tell the CPU where to find the driver routine. There is a strange thing about loading a memory location. You load the least significant bytes (the last two digits) first and then the most significant bytes (the leading two digits).

We are telling the machine that the driver program is located at 127 (7F in hexadecimal, since 7•16 plus 15, F in hex notation being 15, equals 127), 0 (or 00 to complete the four column hex notation). In other words, go to memory location 7F00 and there you will find the driver routine. That was fine when we used the original program, but we've moved it. God only knows what we'll find at 7F00 now!

Remember, we moved the routine up an even 16K to BF00, where we want the CPU to find it. Keeping in mind the backwards loading of the address, POKE 16422 can remain as 0, the last two digits are still 00. Only the 127 will have to be changed. Since BF equals 11•16 (B is hex notation for 11) plus 15, POKE 16423 will now use the value 191.

And, if you happen to look at the data statements, you'll notice a few other 127s. Just as you surmised, they also tell the CPU to go and/or get something from a location in the old 7F range. These too must be changed to 191. If you look over the two disassemblies of the memory locations affected by the two BASIC programs, you'll notice the difference.

We also must change the locations that the data is POKEd into. Line 40 must now read:

```
40 FOR X = -16640 to -16568
```

How, you may ask, can we POKE a value into a negative location? Actually, we don't.

BF00 is memory location 48896 decimal. The trouble is that the TRS-80 does not recognize an integer over 32767 (which, unfortunately, is the end of 16K of memory). To determine the decimal value to POKE into, we take the actual decimal value (48896 for BF00) and subtract 65536 from it (that's the decimal address for 64K of memory + 1.); the resulting negative number is recognized as a valid memory address. However, the MEMORY SIZE__ question should be answered with 48895, one below the actual starting address of the routine.

The same thing would be done if we had moved the routine to the top of 48K of memory (FF00H), incrementing accordingly (instead of 127 or 191 we would use 255, 15•16 plus 15).

Clunk Away Happily, Hopefully

If, after all of this, your printer doesn't clunk happily away, first make sure the TERM/COMM slide switch is really in the COMM position, the baud rate, parity and data bits are set correctly and you've turned on the E/I, then call (800) 433-1679. That's Radio Shack's toll-free Computer Services number. They will fill you in, if they happen to have any data about your specific printer application.

That's most of the pitfalls I've found in using the RS-232-C for printer output. Hopefully, it won't dissuade you from buying it. Only a small investment would then separate you from a modem and then, possibly, time sharing. And the sky's the limit. We might even wind up "talking" one day via our machines. ■

✓Reader Service—see page 226

Find the hidden word.

Puzzler

James P. Morgan
2386-B Ash Creek
Scott AFB, IL 62225

The popularity of word finder puzzles is widespread. They appear daily in most metropolitan newspapers and monthly in a variety of puzzle magazines.

The following program presents one method of creating these popular pastimes with your TRS-80 or another micro with a compatible language structure. The program is written in Level II and requires 16K memory.

Program Operation

After loading from tape and entering RUN, a brief explanation of the game appears on the video. While the user reads this information, the computer forms a 12 × 19 array of random letters and specific words selected from a data bank, inserting them into the array, in random order horizontally, vertically, and diagonally. This process takes 20 to 25 seconds.

Pressing any key clears the screen, and the array with the hidden words is displayed, with the eight selected words listed at the bottom of the array.

The player searches through the jumble of letters for one of the selected words. Once located, he types the word and ENTERs on the keyboard.

The program then prompts for the coordinates, vertical and horizontal, of each letter in the word. As each letter is entered, it is compared for a corresponding match-up on the screen, and if correct, the screen is redrawn with that letter missing. If the match is incorrect, the game ends and the program returns to the beginning.

The first bit of string manipulation begins at line 320. The LEN(string) function extracts the number of letters in a given string and returns a numerical value. This value is assigned to the variable N and used for subsequent string manipulations.

Lines 330 and 340 determine in random fashion in what manner to insert the selected words into the puzzle. For example, if the value of L returns as 1, line 340 sends the program to subroutine 400, the first option in the line.

Assuming the program jumps to line 400, let's follow the logic of just how the program inserts a word into the puzzle from that point.

Subroutines 400 through 1100 all work in the same manner. At line 410, a starting location is selected at random. Lines 420 through 450 then test to find space to insert the selected word.

Line 420 sets up a counter which corresponds to the length of the selected word. The next two lines test to find if: (1) the proposed location is outside the confines of the array; and (2) the proposed location is already in use with another letter.

If the answer to either question is yes, the program goes back to line 410 for another starting point. If all of the proposed spaces are within the array and are unoccupied, the program moves on to insert the word at lines 460 to 490.

The I counter establishes the number of letters to be inserted. The value of the location A$(X – I,Y) is assigned to each letter in turn.

The player continues until he discovers all eight words and removes them from the array. He then opts to continue with a new puzzle or end the game.

Although using the program is quite simple, the programming techniques used to create the puzzle are a bit more complex. I'll present a general outline of the program, followed by a specific discussion of some of the technical aspects in the next paragraphs.

Program Listing 1

```
10 CLS: PRINT@25,"FIND IT":PRINT
20 PRINT"WHILE YOU ARE READING THIS, THE COMPUTER IS BU
   SY
30 PRINT"CHURNING AWAY WITH THOUSANDS OF COMPUTATIONS W
   HICH
40 PRINT"WILL RESULT IN A WORD FINDER pUZZLE. iT TAKES
   ABOUT
50 PRINT"25 SECONDS TO SET UP THE PUZZLE...IT WILL BE C
   OMPLETE
60 PRINT"ANY SECOND NOW. yOU WILL BE SHOWN A SQUARE OF
   JUMBLED
70 PRINT"LETTERS IN WHICH EIGHT LISTED WORDS WILL BE HI
   DDEN.
80 PRINT"LOCATE A WORD, ENTER IT FROM THE KEYBOARD, AND
   THEN
90 PRINT"CONFIRM YOUR LOCATION BY ENTERING THE COORDINA
   TES--
100 PRINT"VERTICAL, THEN HORIZONTAL, OF EACH LETTER IN
   THE WORD.
110 PRINT"TO GET STARTED, TAP ANY KEY."
195 REM *** DIMENSION ARRAYS, POKE FOR READ EXECUTION *
   **
200 CLEAR 500: DIM A$(12,19), C$(50), W$(8):POKE16553,2
   55
205 REM *** SET MAIN ARRAY TO EMPTY SPACE ***
210 FOR X=1 TO 12
220 FOR Y=1 TO 19
230 A$(X,Y)="  "
240 NEXT Y
250 NEXT X
255 REM *** READ IN THE DATA FOUND AT LINE 2000 ***
260 FOR Z=1 TO 50
270 READ C$(Z)
280 NEXT Z
290 A1=0
295 REM *** SELECT WORD, ASSIGN TO W$, PULL LENGTH ***
300 Z=RND (50)
310 W$(A1)=C$(Z)
320 N=LEN(W$(A1))
325 REM *** SELECT INSERTION DIRECTION AT RANDOM ***
330 L=RND(8)
```

✓Reader Service—see page 226

```
340 ON L GOTO 400,500,600,700,800,900,1000,1100
400 REM *** UP VERTICAL SELECT AND PLACE ***
410 X=RND(12): Y=RND(19)
420 FOR I=0 TO N-1
430   IF X-I=0 THEN 410
440   IF A$(X-I,Y)<>"  " THEN 410
450 NEXT I
460 FOR I=0 TO N-1
470   A$(X-I,Y)=MID$(W$(A1),I+1,1)
480 NEXT I: A1=A1+1
490   IF A1=8 THEN 1300 ELSE 300
500 REM *** DOWN VERTICAL SELECT AND PLACE ***
510 X=RND(12): Y=RND(19)
520 FOR I=0 TO N-1
530  IF X+I=13 THEN 510
540  IF A$(X+I,Y) <> "  " THEN 510
550 NEXT I
560 FOR I=0 TO N-1
570   A$(X+I,Y)=MID$(W$(A1),I+1,1)
580 NEXT I: A1=A1+1
590 IF A1=8 THEN 1300 ELSE 300
600 REM *** RIGHT HORZ. SELECT AND PLACE ***
610 X=RND(12): Y=RND(19)
620 FOR I=0 TO N-1
630   IF Y+I=20 THEN 610
640   IF A$(X,Y+I)<>"  " THEN 610
650 NEXT I
660 FOR I=0 TO N-1
670   A$(X,Y+I)=MID$(W$(A1),I+1,1)
680 NEXT I: A1=A1+1
690 IF A1=8 THEN 1300 ELSE 300
700 REM **LEFT HORZ. SELECT AND PLACE ***
710 X=RND(12): Y=RND(19)
720 FOR I=0 TO N-1
730   IF Y-I=0 THEN 710
740   IF A$(X,Y-I)<>"  " THEN 710
750 NEXT I
760 FOR I=0 TO N-1
770   A$(X,Y-I)=MID$(W$(A1),I+1,1)
780 NEXT I: A1=A1+1
790 IF A1=8 THEN 1300 ELSE 300
800 REM *** RIGHT UP D]AGONAL ***
810 X=RND(12): Y=RND(19)
820 FOR I=0 TO N-1
830   IF Y+I=20 THEN 810
840   IF X-I=0 THEN 810
850   IF A$(X-I,Y+I)<>"  " THEN 810
860 NEXT I
```



```
870 FOR I=0 TO N-1
880   A$(X-I,Y+I)=MID$(W$(A1),I+1,1)
890 NEXT I: A1=A1+1: IF A1=8THEN 1300 ELSE 300
900 REM *** LEFT DOWN DIAGONAL ***
910 X=RND(12): Y=RND(19)
920 FOR I=0 TO N-1
930   IF Y-I=0 THEN 910
940   IF X+I=13 THEN 910
950   IF A$(X+I,Y-I)<>"  " THEN 910
960 NEXT I
970 FOR I=0 TO N-1
980   A$(X+I,Y-I)=MID$(W$(A1),I+1,1)
990 NEXT I: A1=A1+1: IF A1=8 THEN 1300 ELSE 300
1000 REM *** LEFT UP DIAGONAL ***
1010 X=RND(12):Y=RND(19)
1020 FOR I=0 TO N-1
1030   IF X-I=0 THEN 1010
1040   IF Y-I=0 THEN 1010
1050   IF A$(X-I,Y-I)<>"  " THEN 1010
1060 NEXT I
1070 FOR I=0 TO N-1
1080   A$(X-I,Y-I)=MID$(W$(A1),I+1,1)
1090 NEXT I: A1=A1+1: IF A1=8 THEN 1300 ELSE 300
1100 REM *** RIGHT DOWN DIAGONAL ***
1110 X=RND(12): Y=RND(19)
1120 FOR I=0 TO N-1
1130   IF X+I=13 THEN 1110
1140   IF Y+I=20 THEN 1110
1150   IF A$(X+I,Y+I)<>"  " THEN 1110
1160 NEXT I
1170 FOR I=0 TO N-1
1180   A$(X+I,Y+I)=MID$(W$(A1),I+1,1)
1190 NEXT I: A1=A1+1: IF A1=8 THEN 1300 ELSE 300
1300 REM *** FILLING IN REST OF A$(X,Y) ARRAY ***
1310 FOR X=1 TO 12
1320 FOR Y=1 TO 19
1330   IF A$(X,Y) <>"  " THEN 1370
1340   B=RND(26)
1350   C=64+B
1360   A$(X,Y)=CHR$(C)
1370 NEXT Y
1380 NEXT X
1390 M$=INKEY$: IF M$="" THEN 1390
1400 GOSUB 1700
1500 REM *** MATCH UP ROUTINE ***
1510 INPUT"ENTER WORD. ENTER '*' TO END.";D$
1520 IF D$="*" THEN 1900
1530 N=LEN(D$):ON ERROR GOTO 3000
1540 FOR I=1 TO N
1550   INPUT"ENTER VERTICAL,HORIZ. COORDINATES OF LETTE
     R MATCH.";X,Y
1560 E$=MID$(D$,I,1)
1570 IF E$<>A$(X,Y)THENPRINT"WRONG!! START OVER":FORX=1
     TO900:NEXT:GOTO10
1580 IF E$=A$(X,Y) THEN A$(X,Y)=" "
1590 GOSUB 1700
1600 NEXT I
1610 GOTO 1500
1700 REM *** THE SCREEN PRINT SUBROUTINE ***
1705 REM ***NEXT LINE, USE 2 SPACES UP TO 9, ONE SPACE
     AFTER ***
1710 CLS:PRINT"1  2  3  4  5  6  7  8  9 10 11 12 13 14
      15 16 17 18 19"
1720 FOR X=1 TO 12
1730 FOR Y=1 TO 19
1740 PRINTA$(X,Y);"  ";
1750 NEXT Y
1760 PRINTX
1770 NEXT X
1780 FOR A1=0 TO 8
1790 PRINTW$(A1);" ";
1800 NEXT A1: RETURN
1900 CLS:PRINT"IF YOU WISH TO PLAY ANOTHER ROUND, TYPE
     'Y' FOR YES."
1910 PRINT"TO END, JUST TAP THE SPACE BAR."
1920 G$=INKEY$: IF G$="" THEN 1920
1930 IF G$="Y"THEN 10
1940 PRINT:PRINT"GOODBY FOR NOW.": END
2000 DATA "COMPUTER","PAPER","TIGER","BYTE","BAUD","HOU
     SE"
2010 DATA "RANDOM","TABLE","GENERATOR","INPUT","OUTPUT"
     ,"OHM"
2020 DATA "RESISTOR","VIDEO","DISPLAY","KEYBOARD","LINE
     ","FOR"
2030 DATA "INPUT","INTERFACE","SIGNAL","MICRO","POWER",
     "SUPPLY"
2040 DATA "SYSTEM","RADIO","WATT","WAIT","STATE","DEVIC
     E"
2050 DATA "PROGRAM","BASIC","CASSETTE","LINEAR","TIMER"
     ,"BYTE"
2060 DATA "PLUS","MINUS","GAME","THEORY","HIGH","LOW","
     BUFFER"
2070 DATA "ELECTRON","AIR","FORCE","NUMBER","BINARY","H
     EX","END"
2995 REM *** ERROR ROUTINE NEXT ***
3000 CLS:PRINT"BAD INPUT. TRY AGAIN.":FORX=1TO1000:NEXT
3010 GOSUB 1700
3020 RESUME 1550


DONE
```

How the Program Works

Lines 10 through 110 print general information concerning the puzzle. As the user reads this information, the program moves on to the computations required to build the structure of the puzzle.

The three arrays used in the puzzle are DIMensioned at line 200. The A$(X,Y) array, used throughout, forms the visual display. The C$(Z) array stores the 50 words, of which eight are selected at random to insert into the A$(X,Y) array. The W$(A1) array stores eight words selected from the C$(Z) array. The POKE statement in this line is required for proper execution of the READ function of the TRS-80, Revision G.

Lines 210 through 250 initially set the 12 × 19 array to a value of empty spaces two columns wide. These empty spaces are eventually filled either by random letters or the specific letters of the puzzle words. Lines 260 through 280 fill the C$(Z) array with the words from the data bank.

Lines 290 through 350 begin the process of selecting the words and inserting them into the A$(X,Y) array. A1 is initially set at 0 and is used as a counter in later subroutines to track the eight selected words. Line 300 selects a random number, which is then used to transfer the word value of that number into the W$(A1) array at line 310.

The letter is derived from the MID$ function which works as follows: the word is identified by the W$(A1) array which was determined at line 310. The specific letter is found by the next two values in the MID$ statement. I + 1 identifies the starting point in the W$(A1) string, while the last value, 1, says we only want one letter extracted.

As the I counter increments, the letters are pulled one by one from the selected word and given a consecutive location in the puzzle array. Line 490 keeps track of the number of words inserted, and when the total reaches eight, the program branches to line 1300. Otherwise, it returns to line 300, selects another word and inserts it in the manner I have just discussed.

Two Loops

Lines 1300 to 1390 complete the job of filling in the entire A$(X,Y) array. Two loops are used to accomplish this task. As the program increments through the Y,X loops, the test at line 1330 checks each array location for occupancy.

If there is already a letter assigned to a specific location, the program jumps to the next value of Y,X. If the location is empty, lines 1340–1360 fill it in with a random letter.

Since the ASCII codes for the letters of the alphabet run from 65 through 90, a random number from 1 through 26, when added to 64, results in a number corresponding to the alphabetical portion of the code. Line 1360 translates that number via CHR$(n), which returns the letter corresponding to the number and assigns it to the A$(X,Y) location.

The array is now complete. It takes about 20-25 seconds for the computer to create the filled-in puzzle, and at this point the user can see the results by hitting any key. The short routine which accomplishes this task is found in line 1390. For those who have not discovered this powerful tool, a word of explanation is in order.

Line 1390 M$ = INKEY$: IF M$ = "" THEN 1390 works as follows: INKEY$ tells the computer to stand by, the user is going to enter something directly from the keyboard. If "" (nothing) is input, the statement tells the program to go back to the beginning of line 1390 and wait for the person at the keyboard to do something other than just sit there.

When any key is pressed, INKEY$ assigns the key value to the string variable M$, and the program moves on to the next line. Simple yet powerful, since it provides the user with instantaneous keyboard control over the program. This function can also be used to advantage with following IF,THEN statements, which can then be used for multiple choice branching related to a specific keyboard input.

Hitting any key moves the program to the GOSUB routine in lines 1700 through 1800. The array and words are printed and control is returned to lines 1500 through 1600.

Line 1510 selects a word and assigns the string value D$. Line 1530 extracts the number of letters in the word which will be used in the counting sequence at line 1540. The error statement at line 1530 is a trap which catches a bad input at line 1550; for example, a keyboard bounce resulting in a coordinate outside the dimensions of the A$(X,Y) array.

The coordinates input at line 1550 are used to select the value of A$(X,Y), which is compared to the value of E$ derived from the MID$ function at line 1560. If the letters match, the A$(X,Y) value is changed to " ", providing positive feedback to the user when the screen is reprinted.

Lines 1900 through 1940 provide the option of ending or going back for another puzzle. Note the use of INKEY$ as a branching device. The DATA is stored starting at line 2000, and the error routine at line 3000.

Modifications and Changes

Everybody loves to play with a program, so I'll offer a few suggestions you may want to try. The number of words in the data bank can be increased to whatever point your available memory handles. This version leaves about 9000 bytes available in my 16K machine, so there is plenty of space for expansion.

The C$(50) dimension at line 200 must be changed to match the number of words in the DATA file, line 210 has to CLEAR more string storage space, and lines 260–300 have to be modified to reflect the number of words available.

The number of words inserted into the puzzle can be increased by setting the value of A1 higher in lines 490, 590, 690, etc. and resetting the loop at line 1780. This modification increases the time required to place all the words into the puzzle.■

Do speed and animation turn you on?

Super Graphics

Alan R. Moyer
993 San Angelo Drive
Hamilton OH 45013

The TRS-80 utilizes various graphics devices; first, SET and RESET (Example 1) and POKE graphics (Example 2). The PRINT method (Example 3) operates faster, allowing simulated animation and other tricks, but you can sit for awhile while all the variables in some programs are initialized into their respective graphics symbols. This PRINT method is commonly called the super graphics method.

```
10 REM SET 1 LINE OF GRAPHICS
20 CLS:FOR X=0 TO
127:SET(1,X):NEXT X
```

Example 1: The SET and RESET graphics function

```
10 REM POKE 1 LINE OF
GRAPHICS
20 CLS:FOR X=15360 TO
15424:POKE X,191:NEXT X
```

Example 2: The POKE graphics function

With the aid of a monitor utility program that allows you to access specific memory locations, it is possible to insert graphics characters into a program line. If you have a disk system, you can call up DEBUG and directly insert graphics characters.

If you don't have a disk system and want to take advantage of super graphics, you can use the program listing, below, to access and change, in decimal or hexadecimal, a BASIC print line to directly print graphics characters under program control without a 'CHR$(n)' initialization. One consideration is that the finished program may not be listable, and you cannot edit any line containing super graphics characters.

Incorporating super graphics into your program with this monitor can be done in two ways. The easiest method is to load the monitor program before you do any programming and then type in the lines as you normally would. The monitor program uses high (65000 and above) line numbers, so it should not interfere with your program input.

If you already have a program and want to add super graphics, you can include the monitor in your program fairly easily by:

1) CLOADing your program (make sure that no line number is higher than 64999);
2) Typing in this program:

```
0 CLS:PRINT"TYPE IN AFTER
MERGING - POKE 16549"PEEK
(16549)":POKE 16548"PEEK(16548):
E=17129
1 S=E:E=PEEK(S+1)*256+
PEEK(S):IF E>0 GOTO 1
2 POKE16549,INT(S/256):POKE
16548,S-INT(S/256)*256:END
```

3) Running the merge program;
4) CLOADing the monitor;
5) Executing in immediate mode the commands listed at the top of the screen.

You should now be able to list the combined programs. You can use the monitor to alter the PRINT statements to include super graphics characters.

The Monitor Program

The monitor program works by typing RUN 65000. Monitor commands include:

D - Set the number base to decimal (for input);
H - Set the number base to hexadecimal (for input);
E xxxxx - examine memory address xxxxx;
M xxx - Modify current memory address to xxx;
; - Steps to the next address (+). Holding the key down continues to step the address until the key is released;
- - Steps to the previous address. Holding the key down continues to step the address until the key is released;
S - Stop program, return to BASIC.

The monitor prints the address and data on the screen in decimal and hexadecimal and the ASCII value (if any) of the address data. You can then modify any address and insert your desired values. The following demonstrates use of the monitor and gives an example of super graphics. This example can be expanded to include elaborate super graphics.

1) CLOAD the super graphics monitor;
2) Type in the following line: '10 PRINT "SAMPLE LINE FOR INSERTING GRAPHICS";
3) Type in RUN 65000;
4) As the monitor takes over, type H to establish hexadecimal number base mode;
5) Type E (for examine), then

```
10 REM PRINT 1 LINE OF
GRAPHICS
20 CLS:PRINT STRING$(64,191)
```

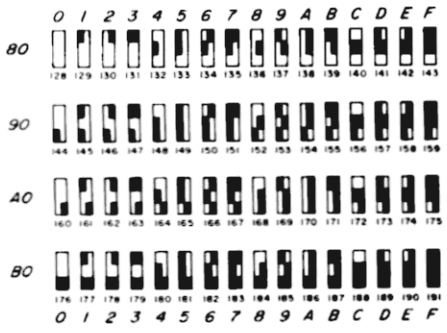
or

```
10 REM PRINT 1 LINE OF
GRAPHICS
20 A$=CHR$(129)+CHR$(131)
+STRING$(3,158)+CHR$(26)
+STRING$(5,24)+CHR$(185)
+STRING$(4,171)
30 REM "26"=LINEFEED, "24"=
BACKSPACE
40 CLS:PRINT A$
```

Example 3: The PRINT graphics function

type in 42E9 (for cassette systems, or 68BA for disk);
6) The monitor displays the address 42E9 and its data. An ASCII value (if any) for the data will be shown in the right hand column;
7) Hold the ; key down to step the address display along. Watch the ASCII column until you see the line you entered show up ("SAMPLE LINE FOR INSERTING GRAPHICS");
8) Use the ; and - keys to get to the beginning of the line;
9) Type in M (modify) and any value for a graphics character (Table 1);
10) The monitor again displays the address with the new data value. Do this as many times as there are characters within the quotation marks of the PRINT statement (when typing in the original program, include enough spaces in your PRINT statement to include all the graphics and cursor control you need);

HEX FORM OF GRAPHICS



1A - LINEFEED
1B - BACKSPACE

11) Type in S to return to BASIC;
12) List the program to inspect your PRINT statement. You should see words like CLS, SET, VAL, etc. They are BASIC commands that the computer normally stores as one byte tokens. These values are normally not available from the keyboard and must be inserted by some other method.

To run the present example, type 20 STOP, so that you can see the display without the monitor program messing up the screen, and run the program. You should now see a number of graphics characters appear on the screen.

You can CSAVE the program with the monitor attached for further development and delete the monitor when finished with a program. ■

Program Listing

```
65000 ON ERROR GOTO 65430
65010 ' UTILITY MONITOR - V2.0 - ALAN R. MOYER 7/3/79
65020 DEFINTB-M,O-S,U-Z:DIMHN$(15)
65030 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
65040 FORX=0TO15:READHN$(X):NEXTX
65050 CLS:LC=64:BA=16
65055 ' PRINT HEADING
65060 PRINT@0,"     UTILITY MONITOR - V2.0      N0. BAS
     E - ";
65070 IFBA=10PRINT"DECIMAL           " ELSE PRINT"HEXID
     ECIMAL      "
65080 PRINT@64,"ADDR.(D)      ADDR.(H)      DATA(D)
       DATA(H)       ASCII";
65090 GOSUB65340
65095 ' GET INPUT
65100 A$=INKEY$
65110 IFA$="D" BA=10:GOTO65060 'CHANGE TO DECIMAL
65120 IFA$="H" BA=16:GOTO65060 'CHANGE TO HEXADECIMAL
65130 IFA$="E" GOTO65190 'EXAMINE AN NEW ADDRESS
65140 IFA$="M" GOTO65290 'MODIFY CURRENT ADDRESS
65150 IFA$="S" THEN ON ERROR GOTO 0: END 'BACK TO BASIC
65160 IFA$="-" AD=AD-1:GOSUB65423:GOSUB65350:POKE16443,
     0:GOTO65240   'DECREMENT ADDRESS BY ONE
65170 IFA$="+" AD=AD+1:GOSUB65423:GOSUB65350:POKE16443,
     0:GOTO65240   'INCREMENT ADDRESS BY ONE
65180 GOTO65100
65185 'GET *EXAMINE* INPUT
65190 ASC@LC,A$;
65200 INPUTA$:GOSUB65340
65210 IFBA=10 AD=VAL(A$):GOSUB65423:GOSUB65350:GOTO6524
     0
65230 H$=A$:GOSUB65400:AD=T:[66NEW5x:GOSUB65423
65235 'GET ADDRESS DATA
65240 D=PEEK(TD):H5=INT(D/16):H6=D-H*16
65245 'PRINT DATA
65250 PRINT@LC,AD;TAB(14)HN$(H1);HN$(H2);HN$(H3);HN$(H4
     );
65260 PRINTTAB(28)D;TAB(42)HN$(H5);HN$(H6);
65270 IF(D>31)FIX(D<192)PRINTTAB(56)CHR$(D) ELSE PRINT
65280 GOTO65060
65285 'GET *MODIFY* INPUT
65290 PRINT@LC,A$;
65300 INPUTA$:GOSUB65340
65310 IFBA=10THEN D=VAL(A$):GOTO65330
65320 H$=A$:GOSUB65400:D=T
65325 'INSERT NEW DATA
65330 POKE TD,D:GOTO65240
65335 'INCREMENT LINE COUNTER
65340 LC=LC+64:IFLC>960 LC=960:RETURN
65345 'GET HEX INvORMATION
65350 H1=INT(AD/4096)
65360 H2=INT((AD-H1*4096)/256)
65370 H3=INT((AD-((H1*4096)+(H2*256)))/16)
65380 H4=AD-((H1*4096)+(H2*256)+(H3*16))
65390 RETURN
65395 'CONVERT HEX INPUT TO DECIMAL
65400 T=0:N=1:FORY=LEN(H$)TO1STEP-1
65410 FORX1=0TO15:IFMID$(H$,Y,1)=HN$(X1) X2=X1:X1=15
65420 NEXTX1:T=T+X2*N:N=N*16:NEXTY:RETURN
65422 'CHECK ADDRESS INPUT
65423 TD=AD
65424 IF AD<0 THEN AD=0: TD=AD: RETURN
65425 IF AD>65535 THEN AD=65535
65426 IF AD>32767 THEN TD=-(65536-AD)
65428 RETURN
65430 PRINT@LC,"ILLEGAL INPUT...TRY AGAIN":RESUME 65060
```

Instant Software™ New Releases

FOR THE TRS-80*

LIFE

Would you like to play god?

Even if you've only been involved with computers for a short while, you're certain to have heard of Life. The game was originally created by British mathematician John Conway and popularized in Martin Gardner's Mathematical Games column in Scientific American magazine. Life, a computerized simulation of the life cycle of a colony of bacteria, allows you to manipulate both the bacteria and their environment.

Over the years the game has lost none of its fascination for computerists. It is based on a few simple concepts but it results in captivating, animated graphics displays.

There are two versions of Life included in this package. The first is written in machine-language and is the most versatile, flexible and the swiftest version of Life we've ever seen. The second is in BASIC with machine-language subroutines. This allows both the machine-language devotee and the BASIC aficionado to experiment with the program.

Patterns can be created and edited easily. You can create your own "creatures" or use the library of preprogrammed creatures. You can run at full speed (100 to 200 generations per minute), enter a pause factor, or single step through the life cycle.

No matter how you approach Life, whether artisitically, mathematically, intuitively, or just for fun, this is THE classic program.

Order No. 0078R $9.95

Winner's Delight

Are you a winner? Do you enjoy challenging yourself with thorny tasks? Then try Winner's Delight. This quartet includes:

Amazing—You must escape from a maze, one that you view from the inside. You must work against the clock—and you may meet a nasty dwarf who can block your passage to freedom.

Junior Checkers—Not your usual game of checkers... The challenge is to beat the computer in the fewest number of moves.

Jumbo Jigsaw—Fit the pieces of the jigsaw together in the fewest number of tries. The program offers three levels of expertise for you to choose from.

Thirteen Ways—Try to fill up your columns with the numbers you roll on dice. Lady Luck may be with you or against you. But you may be certain that the computer will be plotting how to fill its columns first!

You too can be a winner, with Instant Software!

Order No. 0124R $9.95

Body Buddy

Get to know the Inner You. Use this package to learn your caloric needs and to set up a weight-loss diet. It will also introduce you to human anatomy and physiology.

The **Adult Caloric Requirements** program can determine your Basal Metabolic Rate, after you respond to a computerized "questionnaire". Then the program makes recommendations on how you can reach an ideal weight, through dietary planning.

Our **Flexi-Diet** program will create a practical diet for you. Choose your caloric intake, from 600 to 2400 calories per day. The program will make up sample menus for any meal you desire. If you don't care for its choices, it will make as many alternative menus as you like!

In the **Anatomy Quiz** program, a human torso is drawn on your video monitor and you must locate various organs within the body. After you've made your choice, the program gives a mini-lesson, which includes the organ's size, exact location and major bodily functions.

Body Buddy: Let it change you for the better!

Order No. 0109R $9.95

Investor's Paradise

Imagine that you've been given a large sum of money and have the opportunity to see if you can make a killing in the market...

Stock Trek—This is a stock market simulation in which you and up to five other investors buy and sell stocks. See if you can transform $5000 into a fortune in twelve short months. The program has an automatic ticker tape that announces market conditions plus a stock price display board. You can ask for a prospectus that will describe each stock and its dividend potential. Finally, you can see the performance of each stock displayed on a graph. At the end of one (simulated) year, the computer will display the net worth of all investors. The player with the greatest net worth is advised to start looking at the financial pages.

Speculation—This program goes a step beyond being a mere simulation. You enter the financial data on up to 25 real companies and start playing the market. You can buy and sell shares based on net cost, including sales commissions. You'll be able to compare how you did in the market, based on the value of your portfolio and accumulated dividends, versus investing your money at a fixed rate of interest. This program can simulate up to five years of playing the market in computer time and all your data can be stored on tape for future reference. Although this program isn't intended to simulate actual market conditions, it comes darned close.

The Investor's Paradise package lets you experience all the thrills and triumphs of the stock market without risking a dime.

Order No. 0125R $9.95

*A trademark of Tandy Corporation

TO ORDER: Look for these programs at the dealer nearest you (see list on the next page). If your store doesn't stock Instant Software send your order with payment to: Instant Software, Order Dept., Peterborough, N.H. 03458 (Add $1.00 for handling) or call toll-free 1-800-258-5473 (VISA, MC and AMEX accepted).

Prices subject to change without notice.

Instant Software™

Peterborough, N.H. 03458

Ask for Instant Software at a computer store near you.

Alabama

Anderson Computers
3156 University Dr., Huntsville

Computerland of Huntsville
3020 University Dr., Huntsville

Olensky Bros.
3763 Airport Blvd., Mobile

Arizona

Professional Data Systems
4508-A N. 16th St., Phoenix

Millets TV & Radio
621 East Broadway, Mesa

California

AMCO Elect. Supply
635 E. Arrow Hwy., Azusa

Byte Shop
6038 Clairmont Mesa Blvd., San Diego

Byte Shop
123 E. Yorba Linda, Placentia

Byte Shop of Mt. View
1415 West El Camino Real, Mt. View

Byte Shop of Sacramento
6041 Greenback Ln., Citrus Heights

Capital Computer Systems
3398 El Camino Ave., Sacramento

Computers Made Easy
819 East Ave. O-9, Palmdale

Computer Store of San Leandro
701 MacArthur Blvd., San Leandro

Computer World
8791 Westminster Ave., Westminster

Computerland
16720 S. Hawthorne, Lawndale

Computerland of W. LA
6840 La Cienega Blvd., Inglewood

Coast Electronics
3118 No. Main St., Morro Bay

Computerland
24001 via Fabricante No 904, Mission Viejo

Computer Mart of California
315 Diamond Bar Blvd., Diamond Bar

Electronic Systems
4883 Tonino, San Jose

Hobbi-tronics
1378 So. Bascom Ave., San Jose

Hobby World
19511 Business Ctr. Dr., Unit 6, Borthridge

Huntington Computing
2020 Charles St., Corcoran

I.C.E. House Inc.
398 North E. St., San Bernardino

Jade Computer Products
4901 W. Rosecrans, Hawthorne

Malibu Microcomputing
23910A Deville Way, Malibu

Marfam Co.
6351 Almaden Rd., San Jose

Opamp/Technical Books
1033 N. Sycamore Ave., Los Angeles

PC Computers
10166 San Pablo Ave., El Cerrito

Q.I. Computers, Inc.
15818 Hawthorne Blvd., Lawndale

Radio Shack Dealer
8250 Mira Mesa Blvd., San Diego

Radio Shack Dealer
50 N. Cabrillo Hwy., Half Moon Bay

Santa Rosa Computer Center
604 7th St., Santa Rosa

Silver Spur Elect. Comm.
13552 Central Ave., Chino

The Computer Store
820 Broadway, Santa Monica

Colorado

Colorado Computer Systems
311 W. 74th Ave., Westminster

Computerland of North Denver
8749 Wadsworth Blvd., Arvada

Computer Shack
1635 South Prairie, Pueblo

Software Gourmet
1111 S. Pearl St., Denver

The Computer Store
2300 Welton St., Denver

Connecticut

American Business Computers
454 Thames St., Groton

Computerlab
130 Jefferson, New London

Computerland
1700 Post Rd., Fairfield

Computerland
60 Skiff St., Hamden

Computer Works
1439 Post Rd. E., Liberty Plaza, Westport

Instructional Systems Computers
807 Hartford Rd., Manchester

Technology Systems
208 Greenwood Ave., Bethel

D.C.

The Program Store
4200 Wisconsin Ave., N.W., Washington, D.C.

Florida

Al Personal Computer
178 Oxford Rd., Fern Park

AMF Electronics
11146 N. 30th St., Tampa

Boyd-Ebert Corporation
1328 West 15th St., Panama City

Computer Center
6578 Central Ave., St. Petersburg

Computer Junction
5450 So. State Rd. 7, Ft. Lauderdale

Computerland
7374 S. Tamiami Trail, Sarasota

Computerland of Ft. Lauderdale
3983 N. Federal Hwy., Ft. Lauderdale

Computerland of Jacksonville
2777-6 University Blvd. W. Jacksonville

Computerland of Tampa
1520 E. Fowler Ave., Tampa

Computerland of West Palm Beach
4275 Okeechobee Blvd., West Palm Beach

Computer Shack
3336 Beach Blvd., Jacksonville

Computer System Resources Inc.
3222 S.W. 35th Blvd., Gainesville

Curtis Waters Enterprises
236 Talbot Ave., Melbourne

Heath Kit Electronic
4705 W. 16th Ave. Center, Hialeah

HIS Computermation
1295 Cypress Ave., Melbourne

Ukatan Computer Store
Airport Rd., Destin

Williams Radio & TV Inc.
2062 Liberty St., Jacksonville

Your Basic Computer Store
971 Seaway Dr., Ft. Pierce

Georgia

Atlanta Computer Mart
Atlanta

Computerland of Atlanta
2423 Cobb Parkway, Smyrna

Micro Computer Systems
3104 E. Shadowlawn N.E., Atlanta

Hawaii

Computerland of Hawaii
567 N. Federal Hwy., Honolulu

Radio Shack Assoc. Store
1712 S. King St., Honolulu

Idaho

Electronic Specialists
8411 Fairview Ave., Boise

Illinois

Computerland
4507 North Sterling, Peoria

Computerland
9511 N. Milwaukee Ave., Niles

Computer Station
3659 Nameoki Rd., Granite City

Garcia & Associates
203 No. Wabash Ave., Suite 1510, Chicago

Midwest Micro Computers, Inc.
708 S. Main St., Lombard

Indiana

Computer Center of South Bend
51591 US 31 North, South Bend

Data Domain
221 W. Dodds, Bloomington

Fall Creek Electronics Store
732 Center St., Pendleton

Iowa

Memory Bank
1721 Grant St., Bettenbort

Kansas

Central Kansas Computers
6 S. Broadway, Herington

Maine

Maine Computronics
Intown Plaza, Bangor

Mid Maine Computer Co.
158 Turner St., Auburn

Radio Shack
315 Main Mall Rd., So. Portland

Maryland

Computer Age
9433 Georgia Ave., Silver Springs

Jack Fives Electronics
4608 Debilen Circle, Pikesville

The Comm Center
9624 Ft. Meade Rd., Laurel

Massachusetts

ComputerCity
175 Main St., Charlestown

ComputerCity
50 Worcester Rd., Framingham

Computerland of Boston
214 Worcester Rd., Wellesley

Computer Packages Unlimited
342 Boston Turnpike, Shrewsbury

Land of Electronics
1127 Western Ave., Lynn

Lighthouse Computer Software
14 Fall River Ave., Rehobath

Mark Gordon Computers
15 Kenwood St., Cambridge

New England Electronics Co.
679 Highland Ave., Needham

Small Business System Group
Main St., Dunstable

The Computer Store
120 Cambridge St., Burlington

Tufts Radio & Electronics
206 Mystic Ave., Medford

Michigan

Computer Center
28251 Ford Rd., Garden City

Computer Connections
38437 Grand River, Farmington Hills

Computerland of Grand Rapids
2927 28th St. S.E., Kentwood

Computerland of Southfield
29673 Northwestern Hwy., Southfield

Computer Mart
560 W. 14 Mile Rd., Clawson

Computer Room
455 E. Michigan Ave., Kalamazoo

Computronix Corp.
423 S. Saginaw Rd., Midland

Hobby House
1035 W. Territorial Rd., Battle Creek

Main Systems Inc.
1161 No. Ballenger Hwy., Flint

The Alternate Source
1806 Ada, Lansing

The Eight Bit Corner
722 Evanston Ave., Muskegon

TRI Country Electronics & Sound Center
1537 North Leroy, Fenton

Ye Olde Teacher Shoppe
1823 Witmyre St., Ypsilanti

Minnesota

Computerland of Hopkins
11319 Hwy F., Hopkins

Digital Den
Burnsville Center

Minnesota Software Inc.
5422 Fisher St., White Bear Lake

Zim Computers
5717 Xerxes Ave., N. Brooklin Center

Mississippi

Dyer's, Inc.
200 E. Main St., West Point

Softwarehouse
816 Foley St., Jackson

W. Vernon Foster Inc.
816 Foley St., Jackson

Missouri

Century Next Computers
1001 E. Walnut, Columbia

Comp-U-Trs Software Center
51 Florissant Oaks Shopping Center, Florissant

Software Shack
16501 Greenwald Court, Belton

Montana

Intermountain Computer
529 So. 9th St., Livingston

Personal Computer
121 Red Oak Dr., Carl Junction

The Computer Store
1216 16th St. W. #35, Billings

Nebraska

Computerland of Omaha
11031 Elm St., Omaha

Midwest Computer Co. Inc.
8625 I St., Omaha

Midwest Computer Co. Inc.
4442 S. 84th St., Omaha

Midwest Computer Co. Inc.
4403 S. 87th St., Omaha

Scottsbluff Typewriters Inc.
1824 Broadway, Scottsbluff

Nevada

Century 23
4566 Spring Mountain Rd., Las Vegas

New Hampshire

Bitsnbytes Computer Center
568 Pleasant St., Concord

ComputerCity
1525 S. Willow, Manchester

Paul's TV
Main St., Fremont

Portsmouth Computer Center
31 Raynes Ave., Portsmouth

Radio Shack Assoc. Store
Fairbanks Plaza, Keene

Sturdivant and Dunn
124 Washington St., Conway

New Jersey

Abe's TV Sales & Service
College Town Shopping Center, Glassboro

Computer Corner of NJ
439 Rte. #23, Pompton Plains

Computer Encounter
2 Nassau St., Princeton

Computerland
35 Plaza Rte. #4, W. Paramus

Computer Mart of NJ
501 Rte. 27, Iselin

Crowley's
Rd. #3, Whitehouse Station

Dave's Electronics
Pennsville Shopping Ctr., Pennsville

GHB Enterprises Inc.
Rte. 38, Rudderaw Ave., Mapleshade

Lashen Electronics Inc.
21 Broadway, Denville

Personal Computing Inc.
51 Central Sq., Linwood

Radio Shack/J&J Electronic
Mansfield Shopping Ctr.
Rt. 57 Allen Rd., Hackettstown

The Bargain Brothers
Glen Roc Shopping Center
216 Scotch Road, Trenton

The Computer Emporium
Bldg. 103, Avenues of Commerce
2428 Rte. 38, Cherry Hill

New Mexico

Autel Electronics Co.
232 Wisconsin N.E., Albuquerque

South West Computer Center
121 Wyatt Drive, Suite 7, Las Cruces

Thomas E. Carr Jeweler
1300A Tenth St., Alamogordo

New York

Aristo Craft
314 Fifth Ave., NYC

Bertiner Computer Center
102 Jericho Turnpk, New Hyde Park

Bits & Bytes
2800 Straight Rd., Fredonia

Computer Corner
200 Hamilton Ave., White Plains

Computer Era Corp.
1570 3rd Ave., New York

Computer Factory
485 Lexington Ave., NYC

Computer House, Inc.
721 Atlantic Ave., Rochester

Computerland of Nassau
79 Westbury Ave., Carle Place

Computerland of New York City
58 W. 44th St., New York

Computer World
519 Boston Post Rd., Port Chester

Comtek Electronics, Inc.
2666 Coney Island Ave., Brooklyn

Comtek Electronics, Inc.
Staten Island Mall
Store 220A, Staten Island

Digibyte Systems Corp.
31E. 31st St., New York

80-Microcomputer Services
118 Masten Ave., Cohoes

Home Computer Center
671 Monroe Ave., Rochester

Mr. Computer
Imp. Plaza, Rte. 9, Wappingers Falls

Softron Systems
308 Columbia Turnpike, Rensselaer

The Computer Tree Inc.
409 Hooper Rd., Endwell

Upstate Computer Shop
629 French Rd., Campus Plaza, New Hartford

North Carolina

Byte Shop of Raleigh
1213 Hillsborough St., Raleigh

Sound Mill
Siocum Shopping Ctr., Havelock

Ohio

Altair Business Systems, Inc.
5252 North Dixie Dr., Dayton

Astro Video Electronics
504 E. Main St., Lancaster

Cincinnati Computer Store
4816 Interstate Dr., Cincinnati

Computerland
4579 Great Northern Blvd.,
N. Olmstead

Computerland
6429 Busch Blvd., Columbus

Computerland
1288 Som Rd., Mayfield Heights

Computer Store of Toledo
18 Hillwyck Dr., Toledo

Forbees Microsystems Inc.
35 N. Broad, Fairborn

Microcomputer Center
7900 Paragon Rd., Dayton

Micro-Mini Computer World
74 Robinwood, Columbus

21st Century Shop
16 Convention Way, Cincinnati

Universal Amateur Radio, Inc.
1280 Aida Dr., Columbus

Oklahoma

Sounds, Etc.
Hyw. 33, Watonga

Vern Street Products
114 W. Taft St., Sapulpa

Oregon

Computerland of Portland
12020 S.W. Main St., Tigard

Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem

TRS-80 Products Ltd.
3520 S.E. Vineyard Rd., Portland

Pennsylvania

Arico Elect.
302 Wyoming Ave., Kingston

Arico Elect.
Back Mountain Shopping Center,
Shavertown

Audio Mart
518 Fifth Ave., New Brighton

Computer Workshoppe
3848 William Penn Hwy, Monroeville

Computerland of Harrisburg
4644 Carlisle Pike, Mechanicsburg

Computerland of Pittsburgh
5499 William Flynn Hwy., Gibsonia

Erie Computer Co.
2127 West 8th St., Erie

Mighty Byte Computer Center
537 Easton Rd., Horsham

Personal Computer Corp.
24-26 West Lancaster Ave., Paoli

Personal Computer Corp.
Frazer Mall, Lancaster Ave., Frazer

Rhode Island

Computer City
165 Angell St., Providence

Digital World, Inc.
329 Bald Hill Rd., Warwick

South Dakota

CB Radio Shack
21st and Broadway, Yankton

Tennessee

ACS
1100 8th Ave. So., Nashville

Computerlab
671 S. Menden Hall Rd., Memphis

H & H Electronics Inc.
509 N. Jackson St., Tullahoma

Texas

Computer Port
2142 N. Collins, Arlington

Houston Computer Tech
5313 Bissonet, Bellaire

Interactive Computer
7620 Dashwood, Houston

K.A. Elect.
9090 Stemmons Frwy., Dallas

Pan American Elect. Inc.
1117 Conway, Mission

Radio Shack Dealer
21969 Katy Freeway, Katy

The Compute Shop
6353 Camp Bowie Blvd., Ft. Worth

Waghalter Books Inc.
3 Greenway Plaza E., Houston

Utah

DC Computer Co.
1911 West 70 South, Provo

Quality Technology
470 E. 2nd So., Salt Lake City

Virginia

Computer Works
Rte. 6, Box 65A, Harrisonburg

Home Computer Center
2927 Virginia Beach Blvd.
Virginia Beach

Southside Radio Comm.
135 Pickwick Ave., Colonial Heights

Washington

American Mercantile Co. Inc.
2418 1st Ave. S., Seattle

Byte Shop of Bellevue
14701 N.E. 20th St., Bellevue

Computer Connection Inc.
3100 NW Bucklin Hill Rd., Silverdale

Computerland of South King Co.
1500 S. 336 St., Suite 12, Federal Way

Personal Computers
S 104 Freva, Spokane

Ye Old Computer Shop
1301 G. Washington, Richland

West Virginia

The Computer Corner Inc.
22 Beechurst Ave., Morgantown

The Computer Store
Municipal Parking Bldg., Charleston

Wisconsin

Byte Shop Of Milwaukee
6019 West Layton Ave., Greenfield

Computerland
690 S. Whitney Way, Madison

Computerworld
3015 W. Wisconsin Ave., Appleton

Magic Lantern Computed
3313 University Ave., Madison

Petted Microsystems
4265 W. Loomis Rd., Milwaukee

Wyoming

Computer Concepts
1104 Logan Ave., Cheyenne

Puerto Rico

The Microcomputer Store
1568 Ave. Jesus T. Pinero
Caparra Terrace

Canada

CANADIAN DISTRIBUTORS:
Micron Distributing
409 Queen St., W. Toronto, Ont.
M5V 2A5

Instant Software™ 2

PETERBOROUGH, NEW HAMPSHIRE 03458

Displace T-BUG to three different memory locations and you don't need a multiple loader to do it!

Triple Play for T-BUG

W. H. Johnson
1838 Willowhurst
Cleveland, OH 44112

If you purchase T-BUG for your TRS-80, you soon learn that all of the machine language programs you would like to be able to look at are loaded into the same area of memory (42E9-5000). If you haven't been able to invest in some of the multiple loading monitors, such as RSM, which load into 16K, 32K and 48K versions, you are stuck.

This program takes your standard T-BUG monitor, normally loaded from 4380 to 4980, and moves it to three different locations. You can punch a copy of the one that you need or all three.

Block Moves

This program block moves T-BUG into locations 7380-7980, B380-B980 and F380-F980 and corrects all of the address references. If you have a relocator program that fixes all references, it won't work on T-BUG or RSM. Some of the code is altered by single instruction operations that are not apparent, unless you fully disassemble the programs and interpret how they work. This can be easy or difficult depending on the way the programmer writes his code.

To start the program, load T-BUG, using the SYSTEM command, and then use the M command of T-BUG to change the memory locations given in the program. The main program starts at 4C00 and goes to 4CCB. It uses a set of look-up tables located from 4A00 to 4B9F. It also uses a set of high byte variables located from 4BA0 to 4BA6.

The program does three sets of block moves into each of the areas and then goes into a loop which repeats itself three times. The first time through, the table has the values for the 16K version (7380-7980). At the end of the first loop, it goes to two subroutines to change, re-spectively, the high byte of the table address and the variable high byte. At the end of the third loop, the program returns to your normal T-BUG.

Use the J command to jump to address 4C00. When you have done this, you can punch whichever version you want on tape using the P command, as in the following:

```
P 7380 7980 73A0 TBUG73
P B380 B980 B3A0 TBUGB3
P F380 B980 F3A9 TBUGF3
```

Some of the main entry points of T-BUG that save you from writing your own routines are:

```
4380  Entrance from software breakpoint
43A0  Start entrance /17312 decimal
440D  Examine memory/modify routine
4455  Display register contents stored
      in4825-483C
44D3  Punch a system tape
4506  Load a system tape
450F  Display character pointed to
      byIX + 0as 2 hex
4532  Display ASCII character on screen
4589  Get 2 hex characters in A
45A4  Get a hex character in A
45C8  Input a character from keyboard
47DF  Set breakpoint routine
480D  Clear breakpoint routine
```

T-BUG installs a C3 8043 (jump to address 4380) that initializes a new stack, stores all of the registers in 4825-483C, displays the # on the screen,

```
                MAIN PROGRAM
4C00    01 0006         LD BC, NN LENGTH OF MOVE
4C03    11 8073         LD DE, NN START OF NEW AREA
4C06    21 8043         LD HL, NN MOVING FROM OLD AREA
4C09    ED B0           BLOCK MOVE PROGRAM (16K)

4C0B    01 0006         LD BC, NN
4C0E    11 80B3         LD DE, NN
4C11    21 8043         LD HL, NN
4C14    ED B0           BLOCK MOVE 2ND AREA (32K)

4C16    01 0006         LD BC, NN
4C19    11 80F3         LD DE, NN
4C1C    21 8043         LD HL, NN
4C1F    ED B0           BLOCK MOVE 3RD AREA (48K)

4C21    3E 03           LD A,N # LOOPS TO GO THROUGH
4C23    32 A84B         LD (NN),A LOOP COUNTER STORAGE
-------------------------------------------------------
            TABLE ADDRESS,# VALUES, VALUE TO CHANGE
4C26    11 004A         LD DE,NN
4C29    01 0C00         LD BC,NN
4C2C    3A A04B         LD A,(NN) VALUE 1
4C2F    CD 904C         CALL SUBROUTINE TO CHANGE VAL1

4C32    11 164A         LD DE,NN
4C35    01 0700         LD BC,NN
4C38    3A A14B         LD A,(NN) VALUE 2
4C3B    CD 904C         CALL SUBROUTINE TO CHANGE VAL2

4C3E    11 264A         LD DE,NN
4C41    01 4700         LD BC,NN
4C44    3A A24B         LD A,(NN) VALUE 3
4C47    CD 904C         CALL SUBROUTINE TO CHANGE VAL3

4C4A    11 B04A         LD DE,NN
4C4D    01 0A00         LD BC,NN
4C50    3A A34B         LD A,(NN) VALUE 4
```

Program continues

clears the 18 columns and waits for a new command.

You also have to execute the breakpoint before the F command will restore the three bytes to the original program locations. I mention this only in case you decide to write a program with a number of relative jumps and happen to choose the breakpoint within one of the looping structures. Relative jumps use two bytes of code and, of course, can go forwards and backwards.

There is one trick you can use with T-BUG, if you have a 16K machine and are willing to use very short programs in Level II BASIC. You can put the normal T-BUG into memory with the SYSTEM command and then modify the pointers for the BASIC interpreter, so that the computer starts its storage area above the top of T-BUG. Change the following hex locations to 4980 (don't forget to put the LSB first, i.e., 80 49):

```
40F9 80
40FA 49
40FB 80
40FC 49
40FD 80
40FE 49
```

This way, your BASIC starts storing programs at 4980 hex and does not know that T-BUG is in memory. One caution: If you must hit the reset button, the pointers are restored to the normal values of 42E9 in the three locations. If this happens, you get all sorts of gibberish in most cases.

To recover from a crashed program like this, call SYSTEM and, instead of typing the name, use /17312 and press ENTER. You should be back in T-BUG and can reset the addresses using the M command. ■

```
----ADDRESS TABLE 4A00-4BAF----
4A00   9273 0074 8274 B674 0575 1C75 C575 0C78
4A10   2478 0978 AA73 9F73 F773 0574 3274 5E74
4A20   6D74 6D74 6D74 9573 C973 D573 E173 E473
4A30   0A74 0F74 1274 1874 2574 2874 2B74 3674
4A40   3A74 4674 4D74 5274 5274 6374 6A74 7274
4A50   7574 7C74 8774 8C74 9574 9A74 9F74 A774
4A60   AA74 B074 D574 D874 DE74 E474 E774 ED74
4A70   F374 F674 FC75 0875 1C75 2B75 3175 3875
4A80   3B75 3E75 8B75 9075 9375 9775 A675 AB75
4A90   B075 B875 BD75 CE75 DD75 FC75 9276 B176
4AA0   BB76 D576 E177 E477 EA77 0F78 1E74 1E74
4AB0   0275 0B75 0C76 1176 1E76 3176 7976 9776
4AC0   6177 6177 6177 EE73 4576 4876 5476 5D76
4AD0   6A76 7576 7E76 8776 9A76 9E76 A276 C076
4AE0   C576 DF76 F176 FA76 FF76 0777 0F77 1477
4AF0   1877 1B77 2677 2A77 2D77 3277 3877 3E77
4B00   4177 4777 4B77 5A77 6F77 7E77 8777 9277
4B10   9977 AB77 B577 BF77 5177 5177 5177 5177
4B20   8373 8B73 8F73 9973 9C73 AF73 B273 D273
4B30   DA73 E973 1574 1B74 2274 4373 4A74 5874
4B40   AD74 B374 BB74 BF74 D174 DB74 E174 EA74
4B50   F074 F974 FF74 2475 3475 5375 6E75 7175
4B60   7D75 8775 CA75 E775 8E76 A776 B576 CC76
4B70   DC76 E276 E676 3577 3B77 D477 DB77 E777
4B80   ED77 F077 F777 1278 1678 1678 1678 1678
4B90   A273 A773 C373 C373 C373 C373 C373 C373
4BA0   7374 7576 7778 7900 0300 004A 0000 0000
```

Table 1.

```
4C53    CD 904C         CALL SUBROUTINE TO CHANGE VAL4

4C56    11 C64A         LD DE,NN
4C59    01 2B00         LD BC,NN
4C5C    3A A44B         LD A,(NN) VALUE 5
4C5F    CD 904C         CALL SUBROUTINE TO CHANGE VAL5

4C62    11 204B         LD DE,NN
4C65    01 3600         LD BC,NN
4C68    3A A44B         LD A,(NN) VALUE 6
4C6B    CD 904C         CALL SUBROUTINE TO CHANGE VAL6

4C6E    11 904B         LD DE,NN
4C71    01 0400         LD BC,NN
4C74    3A A64B         LD A,(NN) VALUE 7
4C77    CD 904C         CALL SUBROUTINE TO CHANGE VAL7
----END OF LOOP CHECK----
4C7A    3A A84B         LD A,TESTLOCATION
4C7D    D6 01           SUBTRACT 1
4C7F    32 A84B         LD TESTLOCATION,A
4C82    B7              OR A (TEST FOR ZERO)
4C83    00              NOP
4C84    20 03           JUMP RELATIVE IF NOT ZERO
4C86    C3 A043         RETURN TO TBUG AT END
4C89    CD A94C         CALL CHANGE TABLE AND VALUES SUB
4C8C    C3 264C         RE-LOOP IF NOT DONE
4C8F    00              NOP
-------------------------------------------------------
----CHANGE ADDRESS VALUES IN TABLE----
4C90    F5              PUSH AF
4C91    F1              POP AF
4C92    0B              DEC BC (COUNTER)
4C93    ED53 984C       LD (NN),DE SET UP HL ADDRESS
4C97    2A XXXX         LD HL,(NN) (WILL CHANGE EACH PASS)
4C9A    77              LD (HL),A STORE NEW HI BYTE
4C9B    13              INCREMENT DE
4C9C    13              INCREMENT DE
4C9D    F5              PUSH AF
4C9E    00              NOP
----CHECK IF END----
4C9F    78              LD A,B
4CA0    B7              OR A (TEST IF ZERO)
4CA1    20 EE           JR NZ TO 4C91
4CA3    79              LD A,C
4CA4    B7              OR A
4CA5    20 EA           JR NZ TO 4C91
4CA7    F1              POP AF
4CA8    C9              RETURN
-------------------------------------------------------
----CHANGE TABLE PROGRAM----
4CA9    01 C700         LD BC # VALUES
4CAC    21 014A         LD HL START OF TABLE (HI BYTES)
4CAF    0B              DECREMENT COUNTER
4CB0    3E 40           LD A,40
4CB2    86              MOVE HI BYTE UP 16K
4CB3    77              STORE UPDATED HI BYTE
4CB4    23              INC HL
4CB5    23              INC HL
----CHECK FOR END----
4CB6    78              LD A,B
4CB7    B7              OR A
4CB8    20 F5           JR NZ TO 4CAF
4CBA    79              LD A,C
4CBB    B7              OR A
4CBC    20 F1           JR NZ TO 4CAF
----CHANGE VALUE #----
4CBE    01 0007         LD BC,NN
4CC1    21 A04B         LD HL,NN
4CC4    3E 40           LD A,40
4CC6    86              ADD A,(HL) MOVE UP 16K
4CC7    77              LD (HL),A STORE UPDATED VALUE
4CC8    23              INC HL
4CC9    10 F9           DJNZ TO 4CC4
4CCB    C9              RETURN TO 3 LOOP SECTION
-------------------------------------------------------
```

Program Listing 1.

✓Reader Service—see page 226

A slight advance for cassette users.

Take Me Beyond Your Leader

Robert McTernan
42 Aspinwall Road
Red Hook, NY 12571

After trying various brands of short leaderless cassette tapes on my TRS-80 with various degrees of success, I thought I would try some of my Maxell UD cassettes, since I have had nothing but perfect results using that tape for audio applications.

Voila! Again, perfect results. My tape problems disappeared.

Only one problem. Many times, I found myself forgetting to advance the tape manually to the oxide before CSAVEing my programs. Since bits adhere to tape leaders like water to a duck's back, computer time went down the drain. However, I like the idea of using a tape with nonabrasive head cleaning leaders.

You may also have experienced my occasional problem of forgetting, in haste, the little ? after a CLOAD used to verify a program tape. Or how about an unnoticed keybounce when keying your program label? Ever experience any or all of the above?

Two-liner Preventive

Listing 1 is a multi-statement two-liner that prevents all of the above and produces two verified copies of your program.

Place the routine at the end of your program. When you are ready to CSAVE it, type GOTO1000 and off you go. Follow the prompts and you can't go wrong.

Of course, the routine has to be modified to accommodate your CSAVE labels or you may have to use a higher line number. Just be sure that the GOTO address in the PRINT statement, "..... AT 'READY', GOTO1010", is appropriate for the line numbers you use to contain the routine.

The routine advances the leader to about two inches short of reaching the tape oxide. At this point, the CSAVE operation starts and two copies of your program are written. A prompt then advises you to rewind the tape, place the recorder in play and ENTER. The first copy is then verified. Since a CLOAD always returns to BASIC (READY), type GOTO1010 to verify the second copy of your program.

When your program is completed, delete this routine from memory before your final CSAVE.

The concept of the above routine can also be applied to data tapes to snip off your tape leaders. ■

```
1000 CLS:INPUT"READY TAPE & ENTER";X:Z$=STRING$(10,"0"):PRINT#-1,Z$:PRINT"SAVING
 COPY 1":CSAVE"A":PRINT"SAVING COPY 2":CSAVE"B":INPUT"REWIND, 'PLAY' & ENTER";X:
PRINT"VERIFYING COPY 1.  AT 'READY', GOTO1010":CLOADPRINT
1010 PRINT"VERIFYING COPY 2":CLOADPRINT
```

Listing 1.

✓Reader Service—see page 226

Goodies from GALACTIC

Specialty Programs for

TRS-80 Model I - II - III

EDAS 4.0 (Editor/Assembler)
This is the highly acclaimed *"USER ORIENTED"* Assembler for the TRS-80 Model II by GALACTIC. Loaded with features such as assemble to memory, block move, link to debugger, default filenaming, reverse video editing, warm start entry and much more. Now the programmer can write, assemble, test, and debug his code without ever leaving EDAS.
EDAS 4.0 with complete manual (120 pages)
Model II Version ..... Was $229.00
NOW ONLY $179.00

MASS/MAIL SYSTEM
This is the NAME and ADDRESS system for subscription control or large mailing lists. It will handle up to 10,500 records, with a worst access time of less than 15 seconds and usual access of less than one second. All adds, deletes, and edits are instant for the operator and are then completed later in a "batch monitor". Extensive documentation and ongoing support. Requires TRS-80 Model II and 2 disk drives minimum. Contact GALACTIC direct for detailed specifications and prices for your exact needs.
Model II Version ..... Contact GALACTIC for Price

STOCK MARKET MONITOR
This day to day market monitor is designed for the active trader. The system will track the performance of an issue against the market as well as against itself. The package comes with complete documentation and explainations of the formulas that are used by the program. The system is available for the Model I and the Model III TRS-80.
Model I and III cassette version $89.00
Model I and III disk version $99.00

INVENTORY MASTER
Tired of being a slave to an out-of-control inventory? Let GALACTIC'S INVENTORY MASTER put you in control of your inventory. INVENTORY MASTER operates on a TRS-80 Model I and Model III 48K disk system (Minimum of 2 drives with capabilities of up to 4 drives). Drive spanning capabilities allow you to track 2700 inventory items with a 4 drive system (5100 items for the Model III). Unique machine language sort allows for instantaneous item insertion (approx. 15 seconds with 2700 items in system). Item access can be immediate using system-supplied control numbers. Modeled after a proven main-frame system costing tens of thousands of dollars. Complete add/edit/delete capabilities supported. Placement of orders can be machine-generated as well as user-generated, with editing capabilities. Full report-generator included. Exquisitely documented.
Model I Version $159.00
Model III Version $259.00

MODEL II HOST I/O SYSTEM
From the original author of the TRS-80 HOST and TERM systems in the RADIO SHACK "COMMUNICATIONS PACKAGE". This system allows the full control of the HOST facility by your BASIC program. Set the number of nulls to be sent after a C/R, set a command line to be executed if carrier is lost, turn HOST on and off, switch to channel A or B as desired, enable and disable the ability for the remote terminal to "BREAK" BASIC, identify whether a character came from the HOST'S keyboard or from the REMOTE'S and more. No knowledge of assembler needed. All options may be accessed from BASIC or ASSEMBLER. Complete with detailed documentation. Don't isolate your Model II. Let outside terminals access it's computing power.
Model II with TRSDOS 1.2 ..... $179.00
Model II with TRSDOS 2.0 ..... $199.00

MAIL/FILE SYSTEM
This is the name, address, phone number data base manager that has set the standard by which other systems are compared. This system contains advanced editing and output capabilities. The TRS-80 Model I system will handle up to 600 records per file, while the Model III version will handle up to 1150 records and the Model II will handle 2500 records per file. All versions are file compatable and maintain constant sort indexes on both NAME and ZIP CODE. International PHONE numbers and ZIP CODES are supported. Thousands of code combinations are available. The Model II version also has a "word processor" type input editor and fast assembler sorting. Complete documentation is included with each version of MAIL/FILE.
Model I Version ..... $ 99.00
Model III Version $149.00
Model II Version $199.00

ULTRA TREK
This is an all new concept for this type of game, and compares to the others like chess compares to checkers. ULTRA-TREK is a complex, logical game, intended for the serious contestant. It is doubtful that you will ever master this game, but you will certainly enjoy trying! This program requires a TRS-80 Level II, 16K or more. The program is written totally in BASIC and uses 15.5K of RAM.
Model I & Model III Version (cassette only) $14.95

galactic software ltd. ✓254
A Division of GS & WS, Inc.
11520 N. Port Washington Rd.
Mequon, Wisconsin 53092
(414) 241-8030

Money Orders & COD's Shipped Within 24 Hours. Checks allow 2 weeks.

TRS-80 MODEL II ★★ BUGZAP ★★ by SNAPP, INC.

One program to handle all memory and disk sectors
Read and modify all disk sectors
Read and modify all memory
Output disk or memory to printer
Data is displayed in ASCII and Hexadecimal
A MUST FOR SYSTEM AND APPLICATION PROGRAMMERS

ONLY
$50.00

Order Toll Free Today
1-800-543-4628
Ohio Residents Call Collect
513 - 891-5496

SNAPP, INC. ✓232
SOFTWARE DIVISION
8160 CORPORATE PARK DRIVE
CINCINNATI, OHIO 45242

SHIPPED WITHIN 24 HOURS OF RECEIPT OF ORDER
CREDIT CARDS, ADD 3%. IF IN OHIO, ADD 4½% SALES TAX.

Perils in interplanetary space.

Asteroid Adventure

Greg Perry
6104 E. 48th St.
Tulsa, OK 74135

Don Taylor
2855 S. 96th E. Pl.
Tulsa, OK 74129

Despite widely held beliefs, it is possible to write enjoyable programs for the TRS-80 Level II with only 4K of memory. Take Asteroid Adventure for example.

The object of this space game is to guide your ship through the asteroids of space, land on the moon safely and have fuel to spare.

At the beginning of the program, you will be asked to enter your experience level from one to ten. One is for advanced players, and ten is for beginners. The first time you play you might enter a five.

The screen clears and a field of asteroids (*'s) is printed. On the right-hand side of the screen a half-moon appears and in the lower right corner a fuel reading, based on your experience level, is printed.

Your space ship is the greater-than sign in the upper left corner. Use the four arrow keys to guide your ship. Holding a key down causes continuous movement.

The game commences when the first direction key is pressed. At this time, your fuel consumption begins.

Hopefully, you can maneuver your ship to the moon and land safely. On the way, if you hit an asteroid, you will blow-up. Also, keep an eye on your fuel reading; you wouldn't want to be caught in space with no gas!

About the Program

The asteroid field is printed using the random-number generator so the field is different every play. Your ship is moved by using PEEK statements starting at line 190.

Hitting an asteroid or landing on the moon is detected by a PEEK that scans the next screen location in the direction of the current movement.

The speed of the game, as well as the amount of fuel you have are determined by the number that was entered as your experience level. ■

Program Blocks

LINES:	EXPLANATION:
30-115	Instructions
190-220	Input directions for movement
240-270	Landed, blow-up or move
370-410	Draw random asteroid field
420-520	Draw moon
530-540	Blow-up
600-610	Ran out of fuel
620-680	Beginning and experience input

Program Listing.

```
10 CLS
20 GOSUB 620
30 PRINT "INSTRUCTIONS?":P$=""
40 P$=INKEY$: IF P$="" GOTO40
50 IF P$="N" GOTO 120
60 IF P$<>"Y" GOTO 40
70 PRINT " YOU ARE THE CAPTAIN OF A STAR SHIP. YOU HAVE
     TO "
80 PRINT " SAFELY GUIDE YOUR CREW THROUGH THE ASTEROIDS
    TO"
90 PRINT " THE MOON'S SURFACE BEFORE THE FUEL RUNS OUT.
    USE THE"
100 PRINT " ARROW KEYS TO GUIDE YOUR SHIP. HOLDING THE
     KEYS "
110 PRINT " DOWN WILL CAUSE CONTINUOUS MOVEMENT. PRESS
     ENTER TO"
111 PRINT " START."
115 A$=INKEY$:IF A$="" THEN GOTO 115 ELSE GOTO 120
120 P=0:T=(E-1)*10 + 90:N=0
130 CLS
140 GOSUB 390
150 GOSUB 430
160 PRINT @P,">";
170 PRINT@960,"FUEL:";T;
180 REM* INPUT MOVE
190 IF PEEK(14400)=8 THEN AD=-64:N=1
200 IF PEEK(14400)=16THENAD=64:N=1
210 IF PEEK(14400)=64 THEN AD=1:N=1
220 IF PEEK(14400)=32 THEN AD=-1:FG=1:N=1
230 IFN=0 AND AD=0 THEN 190
240 IFP+AD<0 OR P+AD>1023THENAD=0
250 IF  PEEK(15360+AD+P)=42 THEN GOTO 530:REM BLOW-UP
260 IF FG=1 THEN IF PEEK(15360+AD+P)>=129 THEN AD=0:GOT
    O280
270 IF PEEK(15360+AD+P)>=129 THEN GOTO 550:REM MOON
280 PRINT@P,".";:Q=Q+1
290 PRINT@P+AD,">";
300 P=P+AD
310 T=T-1:IFT<=0THENGOTO600ELSEPRINT@965,T;
320 AD=0
330 FG=0
340 REM* SPEED FACTOR
350 FOR X=1 TO E*10:NEXTX
360 GOTO 190
370 REM* SET STAR FIELD
380 RANDOM
390 CR=0:
400 R=RND(10)+2:CR=CR+R:IF CR>1022 THEN RETURN
410 PRINT@ CR,"*";:GOTO 400
```

Program continues

✓Reader Service—see page 226

```
420 REM* DRAW MOON
430 X=63
440 FOR I=1 TO 8
450 PRINT@X,STRING$(I,CHR$(191));
460 X=X+63:NEXT I
470 X=X+1
480 FOR I=8 TO 2 STEP -1
490 PRINT@X,STRING$(I,CHR$(191));
500 X=X+65:NEXT I
510 FOR Y=45 TO 47:FOR X=126 TO 127:SET(X,Y):NEXTX,Y
520 RETURN
530 CLS:PRINTCHR$(23)
540 FORI=1TO150:PRINT@RND(1000),"* B O O M !! *";:NEXTI
    :GOTO690
550 CLS:PRINTCHR$(23):FOR I=1TO10:PRINT@272,"MISSION SU
    CCESFUL":FOR P1=1 TO 50:NEXTP1
560 PRINT@272,CHR$(30):FOR P1=1 TO 50:NEXT P1:NEXT I
570 GOTO 690
600 CLS:PRINT:PRINT:PRINT"SORRY BUT YOU JUST RAN OUT OF
     FUEL (SPACE IS TOUGH ISN'T IT)": FORP6=1 TO900:NE
    XT P6:GOTO 690
610 GOTO690
620 CLS
630 PRINTCHR$(23):PRINT:PRINT:PRINT:PRINT"       ASTEROI
    D ADVENTURE"
640 FORI=1TO1200:NEXTI
650 CLS:PRINT:PRINT:PRINT:PRINT:PRINT"WHAT IS YOUR EXPE
    RIENCE LEVEL"
660 INPUT"<1-ADVANCED TO 10 BEGINNER >";E
670 IFE<1 OR E>10 THEN CLS:PRINT:PRINT:GOTO660
680 RETURN
690 CLS:PRINT:PRINT"DO YOU WANT TO PLAY AGAIN?"
700 A$=""
710 A$=INKEY$:IFA$=""THEN710
720 IFA$="Y"THENRUN
730 IFA$<>"N"THEN710
740 CLS:PRINT CHR$(23):PRINT@210,"THANK YOU"
750 PRINT@336,"FOR PLAYING"
760 FOR L=1  TO 900:NEXT L
770 CLS:FOR I=1 TO 7:PRINTCHR$(23):PRINT@268,"ASTEROID
    ADVENTURE":FOR P1=1 TO100:NEXT P1
775 IF I=7 GOTO 790
780 PRINT@268,CHR$(30):FOR T=1 TO 70:NEXT T:NEXT I
790 FOR I=1 TO 1000:NEXT I:CLS
800 CLS
```


✓Reader Service—see page 226

THE BLINKING CURSOR

MACHINE LANGUAGE ROUTINE

FOR TRS-80* 16K LEVEL II

LOADS IN SECONDS FROM CASSETTE
INTO LESS THAN 1/2K BYTES OF PROTECTED
MEMORY AND TIES INTO ROM KEY SCAN.

Customize Your Cursor:

SELECT ANY CHARACTER OR GRAPHICS BLOCK
AND WATCH IT BLINK!
CHANGE AS OFTEN AS YOU LIKE.

FEATURES:

- FAST REPEATING ← KEY AND SPACE BAR
- INSTANT SWAP WITH RESIDENT CURSOR
- UNAFFECTED BY NEW OR RESET
- WORKS IN COMMAND- EDIT- EXECUTE MODES
- KEY DEBOUNCE

* T.M. TANDY CORP.

SEND $9.95 CHECK OR MONEY ORDER TO:
SEMISOFT 121 MADISON AVENUE NEW YORK, N.Y. 10016

A System Cassette from Semisoft ✓290

GENEALOGY
Compiling Roots and Branches

An extensive family tree system for the Radio Shack TRS-80 64K 1-disk Model II

STORES AND REVIEWS 1000 FAMILY MEMBERS' names, dates, places, marital statuses (2), relationships (father, mother, spouses (2), children (16)), and footnotes (2)

OUTPUTS complete Family Books of info for yourself and others, new and revised pages for Family Book recipients, indices to names and their ID numbers, indented format 8-generation pedigree and descendent charts, lists of dates and footnotes, and blank and filled-in forms to solicit info. Includes comprehensive indexed 70+ page manual.

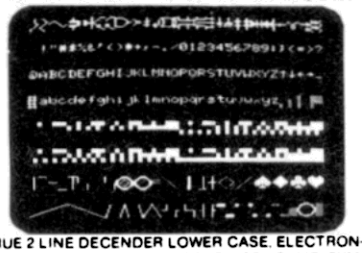
Diskette with manual $250.00
Manual & 9-program 81K BASIC listings $50.00*
Manual alone $25.00*
**applicable toward diskette purchase*

Brief description with example printouts $1.00
Send check or money order or call for more details.

John J. Armstrong
3700 Whispering Pine Rd. #47B ✓414
Mobile, AL 36608
Phone evenings 205/342-7642

From the original author of *Roots and Branches* Personal Computing magazine September 1979
*TRS-80 is a registered trademark of Tandy Corp.

LOWER CASE & GRAPHIC SYMBOLS GENERATOR KIT FOR TRS-80™ CG 16 ..... $64.50



TRUE 2 LINE DECENDER LOWER CASE. ELECTRONIC SYMBOLS. THIN LINE GRAPHICS. GAME SYMBOLS. TEXTURED BACKGROUNDS AND MANY MORE. DEMO CASSETTE IS INCLUDED.
FULLY ASSEMBLED WITH DETAIL INSTRUCTION FOR EASY INSTALLATION.
REQUIRES ELECT. PENCIL TYPE I.C. MOD OR ORDER MEMORY AND SWITCH KIT. SMK. FOR $18.50

SYNCHRONOUS DATA SEPARATOR FOR DISK USERS. SDS $24.50

THE SYNCHRONOUS DATA SEPARATOR WILL ELIMINATE 100% OF THE SOFT READ ERRORS AND SPEED UP DISK ACCESS TIME BY ELIMINATING RETRIES BY THE DISK CONTROLLER.

THE SDS PLUGS INTO THE DISK CONTROLLER'S SOCKET AND HAS ONLY TWO WIRES TO CONNECT. NO TRACES HAVE TO BE CUT. IT IS THE MOST RELIABLE AND ACCURATE DATA SEPARATOR AVAILABLE.

ADD $2.50 FOR S. & H. CALIF. RES. ADD 6% SALE TAX
SEND CHECK OR MONEY ORDER TO:

G.P. ASSOCIATES ✓203
P.O. BOX 22822, SACRAMENTO, CA 95822 (916) 392-0257

TRS-80 IS A TRADE MARK OF TANDY CORP.

HOMES for the TRS-80

✓146

CalData Systems Presents ✓294

WordMagic II

Complete WORD PROCESSING designed specifically for
The Radio Shack TRS-80 Model II Computer

WordMagic II™ is a Word Processor designed specifically for the Radio Shack TRS-80 Model II Computer.*

FEATURES INCLUDE:

- Mailing List/Labels Generation
- Automatic Merging of Mailing Data with Text Files to created "PERSONALIZED" Form Letters
- Automatic wrap-around in text entry
- Margination, Paging, Complete Cursor Movement
- Complete Editing Commands—Insertion, Global Substitution, Overwrite, etc.
- Centering, Smooth Right, Left Justify
- Table of Contents Generation
- Automatic Page Numbering
- Variable Form Lengths
- Underlining
- Line Numbering

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

Requires 64K Model II, TRSDOS & BASIC (not provided with WordMagic)
Introductory Pricing: $195.00 Manual $20. (Cal. Res. add tax)

CalData Systems P.O. Box 178446
San Diego, CA 92117 (714) 272-2661

AT-80 ANNOUNCES A NEWDOS SPECIAL

DDIR80 — Creates program lines of NEWDOS DIR's, adding them to itself. Options include — Search, Re-search, Run, Hardcopy, Display DIR's, Others. Stores up to 175 DIR's. 32K/one disk. $23, w/demo.

CAT — Tic-Tac-Toe with randomly numbered squares. FAST graphics. Human vs human option. $12.

FTDEMO80 — Displays the programs, and the keyboard commands, from the NEWDOS/80 Appendix A examples, *WHILE* executing the programs and commands and displaying results. Cycle through the five file types with only the enter key. $12.

Disk only. Deduct $3 each for 2nd and 3rd program ordered.

NEWDOS $45 NEWDOS+ $95
NEWDOS/80 $145
Add 4% for MasterCard/Visa

AT-80 ✓445
3827 Dismount
Dallas, TX 75211
(214) 339-0498

16K MEMORY EXPANSION KIT FOR YOUR TRS-80, APPLE, AND S-100 COMPUTER

only $59

- 200 Nsec Access, 375 Nsec Cycle
- Burned-in and Fully Tested
- 1 yr. Parts Replacement Guarantee
- Qty. Discounts Available

BETA COMPUTER DEVICES
1230 W. COLLINS AVE.
ORANGE, CA 92668
✓417 (714) 633-7280

80 microcomputing™ bookshelf

3 new books from the editors of KB & 80 Microcomputing

● **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

● **SOME OF THE BEST FROM KILOBAUD/MICROCOMPUTING**—BK7311—A collection of the best articles that have recently appeared in Kilobaud/MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hardcore hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

● **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS** – BK7382 – A valuable addition to your computing library. This two part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of the new microcomputing hobby. Well known authors and well structured text helps the reader get involved in America's fastest growing hobby. $10.95*

Other Books

● **INSIDE LEVEL II**—BK1183—For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

● **TRS-80 DISK AND OTHER MYSTERIES** – BK1181 – by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **PROGRAMMING THE Z-80** – BK1122 – by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **LEARNING LEVEL II** – BK1175 – by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

***Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.**

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing™ bookshelf

GAMES

NEW!

● **MORE BASIC COMPUTER GAMES** – BK1182 – edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

● **WHAT TO DO AFTER YOU HIT RETURN** – BK1071 – PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● **BASIC COMPUTER GAMES** – BK1074 – Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

MONEY-MAKING

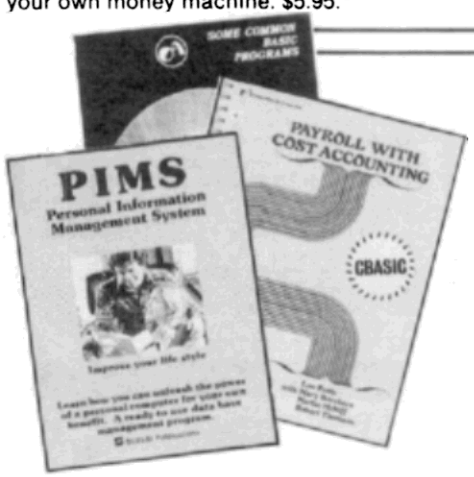
● **HOW TO MAKE MONEY WITH COMPUTERS** – BK1003 – In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY** – BK7306 – According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune – and how you can, too. $2.25.*

● **FREELANCE SOFTWARE PUBLISHING** – BK1179 – by B. J. Korites. "This book is about money and how to make it by writing and selling computer programs," (author's foreword). If you have the skills to write a saleable program, you now need to acquire the skills to sell that program. This compact book comprehensively covers the entire publishing process and many aspects of software salesmanship. $14.95.*

● **THE INCREDIBLE SECRET MONEY MACHINE** – BK1178 – by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

BUSINESS



● **PAYROLL WITH COST ACCOUNTING – IN BASIC** – BK1001 – by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS** – BK1053 – published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $12.50 paperback.*

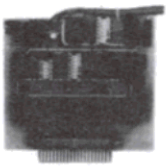
● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM** – BK1009 – Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

TRS-80 SERIAL I/O

• Can input into basic • Can use LLIST and LPRINT to output, or output continuously • RS-232 compatible • Can be used with or without the expansion bus • On board switch selectable baud rates of 110, 150, 300, 600, 1200, 2400, parity or no parity odd or even, 5 to 8 data bits, and 1 or 2 stop bits. D.T.R. line • Requires +5, -12 VDC • Board only $19.95 Part No. 8010, with parts $59.95 Part No. 8010A, assembled $79.95 Part No. 8010 C. No connectors provided, see below.



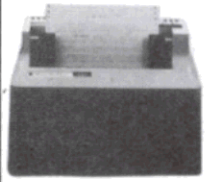
EIA/RS-232 connector Part No. DB25P $6.00, with 9', 8 conductor cable $10.95 Part No. DB25P9.

3' ribbon cable with attached connectors to fit TRS-80 and our serial board $19.95 Part No. 3CAB40.

COMPUCRUISE

$129.95; with cruise control $169.95

PAPER TIGER



Prints address labels, multicopy invoices and legal-size reports. Adjust the tractor width from 1-3/4 to 9-1/2 inches. 8 switch-selectable forms lengths. Print 6 or 8 lines per inch. Add the software-selectable full dot plotting graphics option to print illustrations, block letters, charts, graphs. Part No. 162172 $899.95 • with graphics option Part No. 162173 $1099.95

GAME PADDLES & SOUND

Includes: 2 game paddles, interface, software, speaker, power supply, full documentation including: schematics, theory of operation, and user guide; plus 2 games on cassette (Pong and Starship War). $79.95 Complete Part No. 7922C

DIGICOM DATA PRODUCTS INC. Series 312 Acoustic Coupler

300 BAUD Originate, Part No. AC3122, $219.95. 300 BAUD Answer, Part No. AC3122, $219.95. 300 BAUD Answer/Originate, Part No. AC3123, $229.95.

IBEX LIGHT PEN

Comes with Backgammon and Tic-Tac-Toe on tape with full documentation and program listing. Requires 9v. battery. Part No. IBEX $19.95

SYSTEM EXPANSION from LNW Research

• Serial RS232C/20 mA I/O • Floppy controller • 32K bytes memory • Parallel printer port • Dual cassette port • Real-time clock • Screen printer bus • Onboard power supply • Software compatible • Solder mask, silk screen. PC board and user manual, Part No. LNW80, $69.95.

DISKETTES

Box of 10, 5" $29.95, 8" $39.95. Plastic box, holds 10 diskettes, 5" - $4.50, 8" - $6.50.

16K RAMS

For the Apple, TRS-80 or Pet $8 each Part No. 4116/2117.

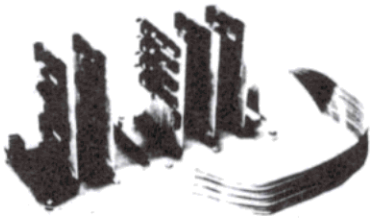
LEEDEX MONITOR

12" Black and White • 12 MHz Bandwidth • Handsome Plastic Case • $139.00

S-100 INTERFACE

AN S-100 bus Adapter—Motherboard for the TRS-80. Kit, Part No. HUH81DLXK, $295.95. Assembeled, Part No. HUH81DLXA, $375.95.

NOW! A FULL SUPPORT SYSTEM FOR TRS-80



• 32K of RAM • EPROM firmware • Disk control • Data acquisition • Parallel I/O • Serial I/O • Plug into GPA's Motherboard. GPA's quality design includes • 6-44 pin edge connectors • +5V, -5V, +12V, -12V external power supply required • Active termination. The Motherboard, Part No. GPA80, is only $149.95.

TAKE ADVANTAGE OF GPA-EXPANSION CARDS FOR THE GPA80

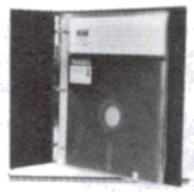
Memory cards: Now with Fortran compilers available for your TRS-80, additional expansion memory is a must! Card with sockets only, Part No. GPA801, $119.95. Card with 16K of 4116 Dynamic Ram, Part No. GPA802, $224.95. Card with 32K of 4116 Dynamic Ram, Part No. GPA803, $329.95. All cards come equipped with sockets to accomodate 32K of Ram.

EPROM firmware card. Put those valuable subroutines in firmware. Don't waste time loading and unloading tapes and disks. For 2708 or 2716 EPROMS, Part No. GPA806, $79.95.

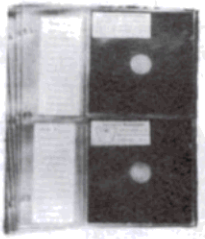
Serial I/O card. Here's what you've been asking for, a full serial terminal interface, with RS-232C or 20 mA. Current loop. Input/output capabilities. Part No. GPA807, $79.95.

Parallel I/O Card. Control functions in the outside world, monitor and store real time events. Two parallel output ports. Dip switches select ports (0-254). Part No. GPA808, $79.95.

FLOPPY DISK STORAGE BINDERS



Three ring binder comes with ten transparent plastic sleeves which accommodate either twenty, five-inch or ten, eight-inch floppy disks. Binder & 10 holders, Part No. 810B—$9.95 • Extra holders, Part No. 810—69¢ each.



Three-ring binder with ten 5 1/4 inch jackets Part No. 510B—$9.95• Jackets only, fits standard 3-ring binders, Part No. 510—69¢ each.

DIGITAL CASSETTE

5 min. each side. Box of 10 $9.95. Part No. C-5.

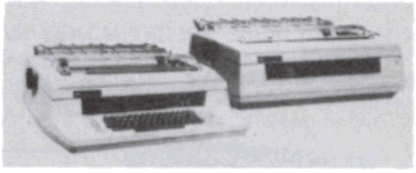
TRENDCOM PRINTER

TRENDCOM 200, Part No. TRC0200 $495.95. **Interface** for TRS-80, Part No. T80A $49.95, For Apple II, Part No. TRCALL, $75.95. For PET, NO. TRCP2, $79.95. For Scoccerer, TRCSR1 $45.95.

SARGON: A Computer Chess Program

Features the complete program that won the 1978 West Coast Computer Faire Tournament. Part No. 00603 — TRS-80 Level II; Part No. 00604 — Apple II (24K). $19.95

SPINWRITER MODELS 5510 and 5520



Features—EIA RS-232C/CCITT V.24 Interface Standard • 55 Characters Per Second Maximum Print Rate • Impeccable Print Quality (OCR Quality) • Microprocessor Electronics • High Resolution Plotting/Graphing • Lowest Operating Noise Level • Self-Test Printing • Operator Engineered Control Panel • Prints Original and up to Seven Copies • NEC Information Systems new Model 5510 Receive Only and Model 5520 Keyboard Send/Receive SPINWRITER terminals are microprocessor controlled serial, impact terminals designed for remote printing applications where impeccable print quality is required. Model 5510 RO, Part No. NECA30759 $2795.95 • Model 5520 KSR, Part No. NECA30762 $3095.95

Send for FREE Catalog...a big self addressed envelope with 80¢ postage gets it fastest!

To Order: Mention part no., description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders (U.S. only) or a VISA or Master Charge no., expiration date, signature and phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 15% for air mail postage and handling. Payment must be in U.S. dollars. Dealer inquiries invited. Prices subject to change without notice.

VISA

26

Order Line: (408) 448-0800

ELECTRONIC SYSTEMS Dept. 80, P.O. Box 21638, San Jose, CA USA 95151

. . . We have them All!

All the Disks and Printers that interface to the TRS-80®

Disk Drive for the TRS-80®

Save over $100 on Radio Shack Price

Percom, Lobo, Vista

Full compatible with expansion interface and TRSDOS software.

OUR PRICE **$369**

***NEW* Digital Innovations Drive** — complete w/case and P.S. Your choice of Shugart or MPI (40 track) drive

A Fantastic Value **$349**

2-Drive Cable. $24.95
4-Drive Cable. $34.95
(Add $6 for shipping and insurance)

CORVUS 10 megabyte hard disk for Mod. I or II
ONLY $4795

ESCON CONVERSION FOR IBM SELECTRIC

Complete w/microprocessor controller and power supply. Factory built. User installs solenoid assembly or it can be done at Escon factory at nominal cost.

Parallel version/TRS-80, $575 **Only $514**
TRS-80 Cable (specify Mod I or II) $25
(Add $6 for shipping)

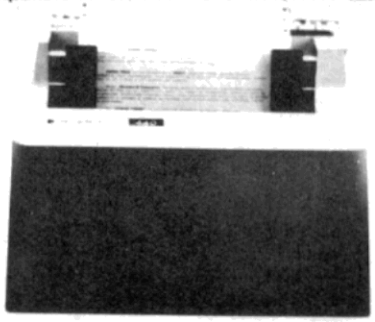
PERCOM
ELECTRIC CRAYON
List $249.95 **ONLY $229.95**

RADIO SHACK® TRS-80®

with standard Radio Shack warranty through Radio Shack stores.

10% off on Mod I and Mod II

Mod I, 4K, List $499 . . . . . **OUR PRICE $449**
Mod I Level 2, 16K, List $849 . . . . . **764**
Mod II w/64K, List $3899 . . . . . **3499**
Expansion Interface with 0 RAM **$269**

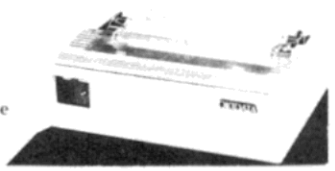


Paper Tiger, List $995 . . $895

w/graphics option, incl. buffer, $1194 . . **$989**
TRS-80 cable . . . . . $ 45
(Add $7.50 for shipping and insurance.)

OKIDATA PRINTER
MICROLINE 80

9x7 dot matrix, 6 or 8 lines per inch, prints 132 columns using compressed print. Prints block formatted graphics as displayed by TRS-80, u/lc, friction feed (prints from roll) or will feed standard 9½" pin-feed fanfold paper. Uses Centronics style parallel interface; plug compatible with all TRS-80's. List $949.



The Microline 80

OUR PRICE ONLY **$649**

Adjustable Tractor Feed Option $99

CENTRONICS PRINTERS

NEW 737
Text Quality Parallel
List $995 **ONLY $849**

NEW 730 w/friction and tractor . . **ONLY $679†**
779 (TRS-80 Line Prtr. I), List $1245 . . . . . **949***
779 w/tractor, List $1350 . . . . . **1049***
702 120 cps, bi-directional, tractor, VFU **1995**
703 180 cps, bi-directional, tractor, VFU . **2395**

*same as Line Printer I †same as Line Prtr. II
(Shipping for 730 and 737 $7.50. Other models shipped freight collect)

NEC SPINWRITER™

A fantastic 55 cps letter-quality printer
A fantastic value . . .**Call for Price!**

TI-810 w/serial/parallel interf. (outperforms Line Printer III).
List $1940 (shipped freight col.) **$1735**

CP/M® OPERATING SYSTEM

for Mod I CP/M 1.4, List $145 . . . . . **$129**
for Mod II CP/M 2.2, List $170 . . . . . **$149**
Shipping $2.50
Complete line of CP/M software available at discount prices — Call us!

TPM — a disk operating system which runs CP/M compatible programs, written specifically for Z80 based systems — more efficient and less expensive than CP/M*
List $79.95 — specify Mod I or Mod II
OUR PRICE **$59.95**

Printers for TRS-80 require Level II machines. Printer cables extra. Call for price and order number.

Prices in this ad are for prepaid orders. Charge cards and C.O.D. 2% higher. Deposit may be required with C.O.D. All prices subject to change and offers subject to withdrawal without notice. Radio Shack and TRS-80 are trademarks of the Tandy Corporation. CP/M is a trademark of Digital Research.

— WRITE FOR FREE CATALOG —

MiniMicroMart, Inc.

24

1618 James Street, Syracuse, NY 13203 (315) 422-4467 TWX 710-541-0431

master charge THE INTERBANK CARD
VISA

Reader Service—see page 226

HOBBYWORLD®
ELECTRONICS, INC.
19511 BUSINESS CTR. DR. DEPT V9
NORTHRIDGE, CALIFORIA 91324
Send For Your Free Flyer Today!

74L$ Order by Cat No. 999 and Type

74L$00	$ .33	74L$74	$ .99	74L$161	$3.75	74L$257	$ .99
74L$01	$ .29	74L$75	$ .59	74L$162	$1.19	74L$258	$ .66
74L$02	$ .49	74L$76	$ .44	74L$163	$1.15	74L$259	$1.99
74L$03	$ .29	74L$83	$ .88	74L$164	$1.15	74L$260	$ .66
74L$04	$ .49	74L$85	$1.15	74L$165	$1.69	74L$261	$2.50
74L$05	$ .35	74L$86	$ .99	74L$166	$3.95	74L$266	$ .66
74L$08	$ .44	74L$90	$ .59	74L$168	$1.29	74L$273	$3.25
74L$09	$ .29	74L$92	$ .75	74L$169	$3.33	74L$275	$4.95
74L$10	$ .44	74L$93	$ .75	74L$170	$2.25	74L$279	$ .49
74L$11	$ .29	74L$95	$ .88	74L$173	$1.25	74L$283	$1.75
74L$12	$ .29	74L$107	$ .55	74L$174	$1.10	74L$293	$1.99
74L$13	$ .55	74L$109	$ .55	74L$175	$ .99	74L$295	$1.99
74L$14	$1.10	74L$112	$ .55	74L$181	$2.50	74L$298	$1.10
74L$15	$ .35	74L$113	$ .55	74L$190	$ .69	74L$324	$1.75
74L$20	$ .29	74L$114	$ .55	74L$191	$1.15	74L$365	$ .99
74L$21	$ .39	74L$122	$ .55	74L$192	$ .99	74L$366	$ .89
74L$22	$ .29	74L$123	$1.15	74L$193	$1.15	74L$367	$ .99
74L$26	$ .77	74L$124	$1.55	74L$194	$1.15	74L$368	$ .99
74L$27	$ .55	74L$125	$ .88	74L$195	$1.15	74L$373	$3.25
74L$28	$ .44	74L$126	$ .88	74L$196	$ .99	74L$374	$4.50
74L$30	$ .39	74L$132	$ .88	74L$197	$1.99	74L$377	$3.25
74L$32	$ .66	74L$138	$1.25	74L$221	$1.99	74L$378	$1.69
74L$33	$ .69	74L$139	$ .99	74L$240	$3.95	74L$386	$ .77
74L$37	$ .39	74L$145	$1.99	74L$241	$2.95	74L$393	$2.25
74L$38	$ .59	74L$147	$2.95	74L$242	$1.95	74L$395	$1.99
74L$40	$ .33	74L$151	$ .88	74L$243	$2.95	74L$399	$2.95
74L$42	$ .88	74L$153	$ .88	74L$244	$3.25	74L$424	$3.25
74L$47	$ .88	74L$154	$3.50	74L$245	$5.00	74L$670	$2.25
74L$48	$ .88	74L$155	$ .99	74L$247	$ .88	81L$95	$1.95
74L$51	$ .39	74L$156	$1.25	74L$248	$ .99	81L$96	$1.95
74L$54	$ .29	74L$157	$1.25	74L$251	$1.99	81L$97	$1.95
74L$55	$ .55	74L$158	$1.49	74L$253	$ .99	81L$98	$1.95
74L$73	$ .44	74L$160	$ .99	74L$256	$2.25		

TTL's Order by Cat No. 999 and Type

7400	$.35	7445	$ .77	74109	$ .55	74176	$ .79
7401	$.35	7446	$ .69	74116	$1.89	74177	$ .77
7402	$.35	7447	$ .66	74120	$ .99	74179	$1.88
7404	$.44	7448	$ .77	74121	$ .44	74180	$ .77
7405	$.44	7450	$ .20	74122	$ .50	74181	$1.88
7406	$.39	7451	$ .50	74123	$ .52	74182	$1.99
7407	$.39	7453	$ .50	74125	$ .52	74184	$1.99
7408	$.35	7454	$ .20	74126	$ .49	74185	$1.99
7409	$.35	7460	$ .29	74132	$ .69	74190	$1.19
7410	$.35	7470	$ .29	74141	$ .77	74191	$1.19
7411	$.39	7472	$ .29	74143	$3.33	74192	$ .77
7412	$.49	7473	$ .36	74145	$7.77	74193	$ .89
7413	$.44	7474	$ .49	74148	$1.29	74195	$ .69
7414	$.66	7475	$ .49	74150	$ .88	74196	$ .88
7416	$.45	7476	$ .38	74151	$ .59	74197	$ .88
7417	$.29	7479	$3.99	74153	$ .69	74198	$1.49
7420	$.35	7480	$ .50	74155	$ .49	74199	$1.49
7422	$.44	7481	$ .99	74156	$ .99	74221	$1.99
7423	$.44	7483	$ .59	74157	$ .63	74251	$ .77
7425	$.38	7485	$ .85	74160	$ .77	74273	$1.10
7426	$.39	7486	$ .35	74161	$ .79	74278	$2.95
7427	$.35	7489	$1.66	74162	$ .79	74279	$ .82
7430	$.35	7490	$ .44	74163	$ .88	74365	$ .69
7432	$.39	7491	$ .59	74164	$ .88	74366	$ .69
7437	$.39	7492	$ .45	74165	$ .88	74367	$ .69
8438	$.39	7493	$ .45	74166	$1.29	74368	$ .69
7440	$.20	7495	$ .65	74170	$1.59	74393	$2.50
7441	$.77	7496	$ .65	74173	$1.09	8T26	$2.50
7442	$.49	74100	$1.69	74174	$ .79	8T97	$2.25
7433	$.69	74107	$ .44	74175	$ .79		

ATARI HOME VIDEO SYSTEM

$183.00

The nation's best selling home video entertainment center is here! Currently supports a library of 32 game cartridges with over 1500 game variations and options. Now you can enjoy all the fun and excitement of an arcade in your own home whenever you wish. Terrific for party entertainment, developing coordination and dexterity, education, or just plain family fun. Comes with interchangeable joystick and paddle controllers, special circuits to protect home T.V., and ATARI'S realistic "combat" game with 108 variations and options. ATARI'S realistic sound effects and crisp, bright colors make the home video center your number one entertainment choice.

CAT NO.	DESCRIPTION	WT	PRICE
2375	ATARI Home Video System	8 lb.	183.00
2206	Driving Controller-Pair	2 lb.	19.95
2207	Paddle Controller-Pair	2 lb.	19.95
2208	Joystick Controller-Pair	2 lb.	19.95

ATARI GAME CARTRIDGES

2376	Starship	6 oz.	19.50
2377	Black Jack	6 oz.	18.75
2378	Space War	6 oz.	12.95
2379	Surround	6 oz.	12.95
2380	Slot Machine	6 oz.	18.75
2381	Outlaw	6 oz.	19.50
2382	Slot Racer	6 oz.	19.50
2383	Video Olympics	6 oz.	19.50
2384	Breakout	6 oz.	19.50
2385	Canyon Bomber	6 oz.	19.50
2386	Street Racer	6 oz.	19.50
2387	Homerun	6 oz.	19.50
2388	Basketball	6 oz.	19.50
2389	Football	6 oz.	19.50
2390	Bowling	6 oz.	19.50
2391	Skydiver	6 oz.	19.50
2392	Fun with Numbers	6 oz.	18.75
2393	Brain Game	6 oz.	19.50
2394	Superman	6 oz.	25.00
2395	Casino	6 oz.	25.00
2396	Backgammon	6 oz.	37.95
2397	Video Chess	6 oz.	37.95
2500	Space Invaders	6 oz.	25.00
2609	Adventure	6 oz.	25.00
2611	Indy 500	6 oz.	37.50
2634	Human Cannon Ball	6 oz.	19.50
2635	Codebreaker	6 oz.	19.50
2636	Flag Capture	6 oz.	19.50
2637	Air-Sea Battle	6 oz.	19.50
2638	Hunt and Score	6 oz.	19.50
2639	Miniature Golf	6 oz.	19.50
2640	Hangman	6 oz.	18.75
2653	Tic Tac Toe	6 oz.	19.50
2654	Circus	6 oz.	19.50

Note: Not for use with ATARI Programmable Computers

ANADEX
DP9500/DP9501 PRINTERS

$1650.00

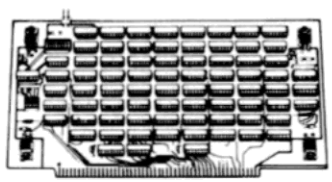
New from Anadex! Two low cost, high performance printers designed for all applications, including standard high-density graphics capability. Both models feature a 9 wire print head with an incredible life expectancy of 650 million printed characters! Full 96 character ASCII set with lower case descenders, double width printing, bi-directional with shortest distance sensing logic. Adjustable-width tractor feed, forms control, horizontal and vertical tabbing, and print up to five copies. Easy interfacing with parallel, RS-232 serial or current loop choices.

The DP9500 is the choice when you require mostly printing and occasional graphics. Select between a 9 x 9 character font and 132 columns, or a 7 x 9 font for 175 columns. Printer speed: 150/200 CPS. Wt. 35 lbs.

The DP9501 is mainly for graphics applications. The 11 x 9 character font produces superb graphics reproduction in 132 columns, and the 7 x 9 character font in 220 columns provides maximum graphics potential. Both models operate at 110VAC, and 220 VAC for European use. Wt 35 lbs.

Cat No. 2551 DP9500 printer
Cat No. 2552 DP9501 printer

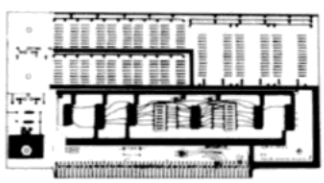
SSM MB6B
8K STATIC RAM BOARD $135



8K bytes by 8 bits, fully buffered, compatible with 8080, 8085, and Z80. Dip switch addressing of independant 4K halves lets the MB6B think like two 4K boards, or one 8K board. Independent 4K addressing allows the flexibility to meet varying software memory needs. Uses low power 21L02 RAM's, operates at 2 or 4MHZ, and is compatible with direct memory access controllers.

Cat No.	Description	Price
*1400-A	450ns kit	$135.00
*1400-B	250ns kit	$147.50
1401-A	450ns a & t	$209.00
1401-B	250ns a & t	$225.00
*1402	Bareboard	$ 23.75

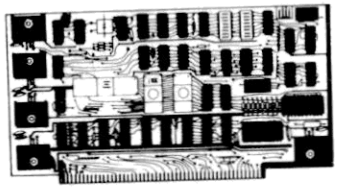
SSM OB1
VECTOR JUMP & PROTOTYPING CARD
$41.25*



Plug compatible for S-100 bus systems, features full 16 bit vector jump address with dip selection of 8080 or Z80. Can be set to jump on power-on-clear, reset, or both. Prototyping areas on the card for ten 16-pin IC's three 24-28 pin IC's and two spare regulator patterns.

*Cat No. 1429 OB1 kit	$41.25
Cat No. 1430 OB1 a & t	$85.00
Cat No. 1431 OB1 bareboard	$32.00

Call Toll-Free: USA (800) 423-5387
In California: (800) 382-3651
Local & Outside USA: (213) 886-9200



SSM CB1-A
8080 CPU BOARD

Just add an I/O board and it's a computer! 256 bytes of on board RAM, with option for 2K of on board PROM. Includes a power-on, preset jump circuit, and MWRITE is available, allowing use without a front panel. There's a parallel input port with status, and AIP controlled addressing: or PROM in 2K blocks, vector jump in 2K increments; RAM in 256 byte increments; RAM in 256 byte increments; input port for addresses 0 – 31 in decimal.

$159 kit

*Cat No. 1403	CB1-A kit	$159.00
*Cat No. 1441	CB1-A bareboard	$28.75

*** Denotes excess inventory sale. No further discounts shall apply**

HAYDEN **SARGON II**

• *Available in Cassette or Diskette*

Sargon II has seven levels of play! When setting up the board the user can scan up and down, left and right before choosing a move. The computer displays the level in which it is thinking, shows the moves that it is contemplating, and then displays the move it has chosen. It comes with a randomized book of opening moves for all 7 levels of play, and a special hint mode that will suggest moves, for players who may need some help.

Cat No.	Description	Weight	Price
2082	TRS-80, L2 16K Cassette	6 oz.	$29.95
2083	Apple II, 24K Cassette	6 oz.	$29.95
2352	TRS-80, L2 16K Diskette	6 oz.	$35.95
2342	Apple II, 24K Diskette	6 oz.	$35.95

INVENTORY '5'

Allows the inclusion of alphabetic information and a data index code in the form of data statements within the program. Includes: **1) Reports;** User specifies up to three numeric, and either or both alpha informations to be listed. Can be vendor specific. **2) Cost/value summary;** Searches all stock areas and reports cost/value quantity, total value by line item, and grand total. **3) Recorder search;** Compares current stock level against specified reorder point, and displays all line items in need of reorder, along with tentative reorder information. **4) Index;** Uses arbitrary file numbers reflecting the order in which the data codes are stored. Reveals file names and numbers in groups of 24. **5) Detailed Rept;** Stock files can be called by file number to reveal memory information. **6) Read and write file;** Stores and reenters data from day to day. **7) Data change;** Updates Data Base.

Cat No. 2056 TRS-80 L2, 16K, cassette **$24.95**

MICROSOFT
TRS-80 FORTRAN PLUS

Create your own subroutines, assembler source files, data files, and FORTRAN files. Fully compatible with TRSDOS, the FORTRAN compiler can operate 1200 lines per minute on a single pass. Also generates a fully symbolic listing of the machine language generated. Macro assembler accepts Z80 opcodes and supports complete Intel standard macro facility. Linking loader provides a variety of capabilities, executed by means of easy command lines and switches. Any number of programs may be loaded with just one command. Text editor random access, line oriented. Only the library routines required to run a particular FORTRAN program need be loaded before execution. Non-standard I/O drivers for each Logic Unit*Number may be written, simplifying the task of interfacing non-standard devices to TRS-80 FORTRAN programs.

Cat No. 1341 TRS-80 L2, 32K w/disk **$185**

PROGRAMMA
TIELINE
$24.95

The most complete communications for the TRS-80, featuring: Host computer: Turns your TRS-80 into a timeshare mainframe. Smart Terminal: Operate another TRS-80 or mainframe timeshare system from your TRS-80. Send/receive BASIC programs and data. Allows exchange of programs over phone lines (modem required). Operate a serial printer, test your modem and RS232, and much more! Complete with extensive manual on telephone communications.

Cat No. 2137 TRS-80 L2, 32K modem, DOS 2.1, RS232 cassette, used w/disk

Introduction to
TRS-80 GRAPHICS

Bob Albrect and Don Inman

One of the most outstanding capabilities of the small computer is the ability to provide graphical data displays that formerly cost thousands of dollars. This book provides a basic introduction to graphic programming using dozens of real examples which run on the Radio Shack TRS-80 Computer. The book begins with basic concept line drawing, then leads the reader to more complexed geometric shapes, moving figure animation, and other more advanced topics. No mathmatics is required, but an understanding of BASIC language is assumed. A TRS-80 Computer for running examples is recommended, but concepts will apply to most low cost computers with graphics capabilities. (175 pages)

Cat No. 2544 **$8.95**

PROGRAMMA
DATA BASE MANAGEMENT 5
$49.95

An easy method of creating data files and storing them in disk memory for future use. Allows you to store and manipulate data for maximum productivity, and modify or incorporate your own routines. Uses Radio Shack's TRSDOS/BASIC language.

Cat No. 2146 TRS-80 L2, 16K

TRS-80 ELECTRIC PENCIL

Allows you to produce mailing lists, forms, large numbers of original correspondence, etc. A character-oriented word processing system, providing maximum freedom and simplicity in the handling of text. Eliminates the need for word hyphenations or carriage returns. Line formatting is done automatically. Insert, delete, or relocate any text using simple keyboard commands.

Cat No. 1338	TRS-80, L1 & L2, 16K, cassette	6 oz.	$ 95.00
Cat No. 1338-D	TRS-80, L1 & L2, 16K, disk	6 oz.	$145.00

TRS-80 CP/M
$149.95

A file-oriented disk operating system that provides a common set of utilities for program development and operation. There are six built-in commands, plus utilities called in from disk. Runs on as little as 16K of memory and one disk drive. Complete with six manuals. CP/M is a registered trademark of Digital research.

Cat No. 1679 TRS-80 L2, 16K w/disk

VERBATIM 525 SERIES
5¼" Diskettes

- *Double Density*
- *Single sided*
- *Perfect for commercial and general applications*

Cat No.	Description	Type	Use for	10 for
1147	Soft sector	525-01	TRS-80	$33.00
1148	10 hole, hard	525-10	North Star, Apple	$33.00
1149	16 hole, hard	525-16	Micropolis	$33.00

VERBATIM 577 SERIES
10 PER BOX

- *Certified twice, 77 tracks*
- *Single sided, double sided*
- *Built-in hub protector ring*
- *For critical data applications*

2330	Soft sector	577-01	TRS-80, etc	$49.95
2331	10 hole, hard	577-10	North Star, Apple	$54.95
2332	16 hole, hard	577-16	Micropolis, etc	$49.95

DISK/DISKETTE DRIVE
HEAD CLEANING KIT

Diskette drive heads require periodic maintenance to assure efficient and error-free operation. Unlike other peripheral devices, the read/write head(s) on disk drives are extremely difficult to clean without partially disassembling the unit. But now, with Hobbyworld's disk drive head cleaning kit, the user can clean these hard-to-reach heads in just minutes! Available for both 5¼ and 8" drives, single and double sided. Comes complete with two cleaning disks, 4 oz of CS-85 cleaning solution, and easy-pour dispenser. Wt 12 oz.

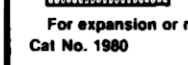
Cat No. 2499 8" disk **$30**
Cat No. 2534 5¼" diskette

TRS-80 and APPLE
16K MEMORY ADD-ON
$55.00

Everything you need to upgrade your system! Includes 4 pages of illustrated instructions. Complete with RAM's and pre-programmed jumpers. No special tools required! Wt. 4 oz.

Cat No	Description
1156	For TRS-80 Keyboard Unit
1156A	For TRS-80 Exp. Interface purchased before 4/1/79
1156B	For TRS-80 Exp. Interface purchased after 4/1/79
1156C	Apple II
1156D	EXIDY

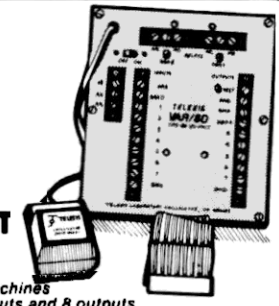
TRS-80 EDGE CONNECTOR



- *40 contacts (2 x 20)*
- *.01" spacing*

For expansion or main unit. Wt. 4 oz. **$4.75**
Cat No. 1980 2 for $9

TELESIS VAR/80
TRS-80 I/O UNIT
$105



- *For level II machines*
- *Provides 8 inputs and 8 outputs*
- *Plugs directly into your TRS-80*

Now you can use your TRS-80 as a digital door lock, burglar alarm, power manager, frequency counter, light dimmer, darkroom timer, just to name a few! Comes fully assembled and tested. Use it with or without the expansion interface! Data pack includes instructions, applications, sample circuits, and several programs. With power supply. Wt. 5 lbs.

Cat No. 1092

SEND FOR FREE FLYER FEATURING . . .

Page after page of impressive, state-of-the-art products. Included in flyer are computers, terminals, disk-drives, printers and many more electronics peripherals that can add dimension to your personal computing. We also stock computerized games and toys, application boards, a large selection of comprehensive software, electronics books, electronics parts, integrated circuits, P.C. & soldering accessories and much, much more.

For all your electronics' or hobbyist's needs, circle our Reader Service number or write/phone us for your free illustrated flyer today!

HOW TO ORDER

Minimum Order $15.00. Order by phone, mail, or at our retail stores. Pay by check, Mastercharge, Visa or C.O.D. Please include expiration date with charge card orders. U.S. $$ only. Include phone number and magazine issue you are ordering from. Add $1.25 for C.O.D. and shipping charges from rates below.

Shipping Rates: U.S.A.
Ground: Add $2.25 for first 2 lbs. and 40c each addt'l. lb.
Air: Add $3.25 for first 2 lbs. and 70c each addt'l. lb.

Shipping Rates: Foreign
Ground: Add $3.00 for first 2 lbs. and 60c each addt'l. lb.
Air: Add $11.25 for first 2 lbs. and $5.00 each addt'l. lb.

Prices Valid through month of issue. Not responsible for typographical errors. Some items subject to prior sale or quantity limits. 120 day guaranteed satisfaction.
Exception: Partially assembled kits, abuse or misuse.

HOBBYWORLD® ELECTRONICS INC.

19511 BUSINESS CTR. DR. DEPT V9 NORTHRIDGE, CALIF 91324

↙23

↙Reader Service—see page 226

THE TOOL BOX

DISTRIBUTORS OF TOP QUALITY DISKS, DOS & DATA MANAGEMENT SYSTEMS

PROGRAMMING TOOLS

SAVE Any 3, $49.95
For Model II $74.95

TDAM $19.95
For Model II $29.95
Includes MTC QUE Card!

Having trouble with RANDOM FILES? With MTC's Table-Driven Access Method (TDAM) you'll never fret over FIELDing again. No knowledge of random access files is required. Insert the TDAM "interpreter" into any BASIC program and type in a few DATA statements describing the information in your files. TDAM does the rest! Reads and writes fields and records of any type (even compresses a DATE field into 3 bytes!). Features automatic file buffer allocation/deallocation, memory buffering, sub-record blocking/deblocking, and handles up to 255 fields per record. Super fast and super simple! Complete with TDAM interpreter, instructions and demo program. Requires programming experience.

DIVERGE $19.95
For Model II $29.95
Compares two BASIC program files, showing the differences between them. Identifies & lists lines which have been inserted, deleted, & replaced. Use for version control.

REBUILD $19.95
For Model II $29.95
Reorganize programs for adding program code, faster execution, readability. Much more than simple renumbering. Rearrange groups of statements within a program - automatically updates references to line numbers. Use with SUPERSEDE and MINGLE for maximum effect.

SIFTER $19.95
For Model II $29.95
Twelve in-memory high-speed sorts for use in any BASIC program: stable, non-stable, with/without tags, for numeric or string data. Random File Sort included. Some sorts written in machine code. Includes sort subroutines, demo programs and instructions. Relocate as needed with REBUILD. Requires programming experience.

SHRINK $19.95
For Model II $29.95
Makes Every Byte Count! Make programs smaller and faster! Combines lines & removes unnecessary code including remarks, without altering program operation. Typically reduces program size 25% to 40%.

SUPERSEDE $19.95
For Model II $29.95
A "must have" for the professional programmer or the serious amateur. Probably one of the greatest time-savers available. Write programs in shorthand - change variable names - generate program documentation - use with REBUILD and MINGLE to build new programs from old ones.

MINGLE-II $19.95
For Model II $29.95
Merge up to 14 files (Program or Data) into a single file. Data files may be merged in ascending or descending sequence with the ordering based on a user-specified comparison field. A very handy utility for consolidating data files.

Single sided, Single density, Soft-sectored

DISKETTES

Verbatim 5¼-inch

$24.95 Box of 10

Quantity 10 Boxes $23.50
Hard-sectored (10-hole), Box of 10 ..... $26.95
8-inch FLOPPIES
Single-density, Box of 10 ..... $29.95
Double-density, Box of 10 ..... $39.95
PLASTIC LIBRARY CASES
5¼-inch or 8-inch diskette case $3.00

FACTORY FRESH, ABSOLUTELY FIRST QUALITY
Minimum order 1 box NO order limit!

Complete for Model I with all utilities
Plus exclusive MTC QUE card!

NEWDOS + $69.95 by Apparat

40 TRACK VERSION ..... $ 79.95

includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

MTC QUE Card only $1.50

Let Your TRS-80® Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 ..... $29.95

Let your TRS-80® Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80® Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 ..... $69.95

The perfect supplement for your NEWDOS+, from IJG, Inc.

"TRS-80 DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

132 pages written in PLAIN ENGLISH packed with HOW TO information with details, examples and in-depth explanations. Recover lost files and directories, remove file protection, make BASIC programs unlistable. How to use SUPERZAP, recover from DOS errors and MORE!

TRS-80 DISK $19.95

All products guaranteed for replacement only. Prices, Specifications & Offerings subject to change without notice.

* MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

* QUANTITY DISCOUNT INQUIRIES INVITED

* WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

LTM INCORPORATED ✓450

TO ORDER CALL TOLL FREE
1-800-321-2037
IN OHIO CALL COLLECT (216) 566-9130

TRS-80 and Radio Shack are registered trademarks of Tandy Corp

THE CHESTERFIELD BUILDING • 1801 EAST 12TH STREET, SUITE 222 • CLEVELAND, OHIO 44115

✓Reader Service—see page 226

ADVERTISERS

*This advertiser prefers to be contacted directly.

When It Comes To TRS-80 Add-on Memory...

LOBO Has It All.

LOBO DRIVES manufactures disk drive subsystems designed to provide TRS-80* users with a wide selection of low-cost, high-speed, efficient, mass-storge capabilities. Every LOBO DRIVES Memory System is thoroughly tested and burned-in to assure reliability and carries LOBO's unique one year, 100% parts/labor warranty.

Expansion and enhanced capabilities are key words in achieving full utilization of your computer system. LOBO DRIVES complete line of TRS-80 compatible disk drive subsystems is the ideal, cost effective way to provide the expansion capabilities you need to meet your system growth requirements.

TRS-80 is a trademark of Radio Shack, A Tandy Company

TRS-80 MODEL II

LOBO DRIVES makes expanding your TRS-80 Model II very, very easy. Now you can add more floppy disk memory at less cost. And, LOBO can provide you with up to 40 MBytes of fixed disk Winchester technology storage capacity that is completely software compatible to your Model II.

- Model 800-850 8-inch dual Floppy Systems
- Model 1850 Dual Floppy/Fixed Disk Memory System

MODEL 1850 DUAL FIXED/FLOPPY DISK MEMORY SYSTEM

LOBO DRIVES has combined a 5 or 10 MByte Winchester technology fixed disk and 1.6 MByte double-sided, double-density floppy disk drive in one cabinet. The unique controller can accommodate two dual units. Now you can have the speed and reliability of fixed disk, with built-in floppy back-up.

- 5 or 10 MByte Fixed Disk Capacity
- Up to 1.6 MByte Floppy Disk Capacity
- Winchester Reliability
- Software Compatible

MODEL 800/850 DUAL FLOPPY DISK MEMORY SYSTEM

Complete with stylized cabinet, power supply, controller, interface, and cables, the Model 800/850 Dual Floppy Disk Memory System is the ideal way for the serious user to expand his disk-based TRS-80.

- Up to 3.2 MBytes Capacity
- Single-side, Single or Double Density
- Double-Side, Single or Double Density
- Complete Software Compatibility
- High Speed Access Time

MODEL 400 5¼-INCH FLOPPY DISK MEMORY SYSTEM

A low-cost, high performance, software-compatible Floppy Disk for TRS-80 Model I users.

- Up to 220 KBytes Capacity
- Single/Double Density
- Soft Sector Format
- 298 Msec Access Time

MODEL LX80 EXPANSION INTERFACE

LOBO DRIVE's new Model LX80 expansion interface enhances system performance by expanding disk storage capacities beyond 40 MBytes, adding a second serial port and facilities for an additional 32 K RAM. The LX80 permits you to achieve the maximum expansion capabilities of your TRS-80.

- Connects Directly to Keyboard
- Two Serial Ports (optional)
- One Parallel Expansion Port (standard)
- One Parallel "Centronics" Printer Port (Standard)
- Supports Double Density 5¼ and 8 inch Floppies
- Separate Port for 8-inch Floppies
- Switch for Overriding Keyboard ROM
- Separate Port for Fixed Disk Drives

MODEL 950 DUAL FLOPPY/FIXED DISK MEMORY SYSTEM

LOBO combines the outstanding capabilities of the latest technological breakthrough in disk drives, the Shugart Technology 5¼-inch Micro Winchester fixed disk drive with the proven reliability of the Model 400/450 Floppy Disk in one easy-to-use cabinet.

- The Storage Capacity of 16 double-sided, double-density Mini-Floppies
- Built-in Floppy Disk Back-up
- 170 Msec Average Access Time
- Sealed Environment/Winchester Reliability

NOTE: Limited Availability in the Fall, 1980

See your nearest dealer, call, or write for the complete LOBO DRIVES story... find out just how competitively priced a quality drive can be.

✓15

LOBO drives INTERNATIONAL

935 Camino Del Sur
Goleta, California 93017
(805) 685-4546
Telex: 658 482

STARTER KIT

EXATRON STRINGY FLOPPY FOR THE TRS-80

Recommended initial purchase:

Exatron Stringy Floppy	$249.50
3 Wafers each: 5', 10', 20', 50'	40.00
Bus Extender, 2-for-1	15.00
ESF Machine Language Monitor	9.95
Wafer Organizer	5.00
	$319.45

SPECIAL PRICE FOR THIS STARTER KIT	$299.50
Sales Tax (California only)	
Shipping and Handling	5.50
TOTAL	

For more information see the current Exatron Stringy Floppy Owners Association Newsletter in *Microcomputing*.

If you have any questions about the product, about Exatron, or ESFOA, please call the Hot Line. Address letters to ESFOA, 3559 Ryder St., Santa Clara, CA 95051.

Stringy Floppy is a trademark of Exatron Corporation.

HOT LINE (For Calls Outside CA) 800-538-8559

v3

exatron

excellence in electronics

3555 Ryder Street • Santa Clara, California 95051
(408) 737-7111